R documentation

of ‘C:/Users/rvanraap/ownCloud/F’ etc.

July 19, 2019

R topics documented:

`addPalette` *Manually add a heatmap palette and view the currently loaded palettes.*

Description

`addPalette()`: function to manually add a colorpalette for heatmaps to the six heatmap palettes existing inside bactMAP. Use `showCurrentPalettes()` to print mockup-heatmap plots with the six default palettes and - if appliccable - the added palettes.

Usage

```
addPalette(palList = list("colorhex1", "colorhex2", "colorhex3")), palName = "YourPalette")

showCurrentPalettes()
```

Arguments

`palList`	a list consisting of 3 hex color codes (character strings, for instance "#FFFFFF" or "#DD0088") or color names known by R (for instance "red" or "violet"). Optional; if argument is not used, the function will ask for the colors seperately while running.
`palName`	a character string defining the name of the added colorpalette. Optional, if argument not used, the function will ask for the palette name.

Value

`addPalette()` and `showCurrentPalettes()` will print heatmap plots using the color palettes loaded into bactMAP. `addPalette()` also adds the manually entered color palette to the list of color palettes loaded into the bactMAP environment during the R session.

Author(s)

Renske van Raaphorst

References

The default color palettes are chosen based on the Color Universal Design (CUD, Masataka Okabe & Kei Ito): `http://jfly.iam.u-tokyo.ac.jp/color/`

See Also

For more information and vast choice of good color palettes for plots, check `http://colorbrewer2.org`

For a comprehensive list of R color names, check `http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf`

Examples

```
##Check the color palettes inside bactMAP
showCurrentPalettes()

##Pick three colors you like and put them in a list - in the order of lowest value - midpoint - highest value
listofcolors <- list("orangered", "darkseagreen", "burlywood1")

##Add to the color palettes inside bactMAP
addPalette(listofcolors, "MyBeautifulPalette")
```

---

bactKymo *Creates kymographs or demographs from pixel values & cell dimensions.*

Description

bactKymo() takes the output of `extr_OriginalCells()` and makes kymographs or demographs of them, depending on the chosen options. By default, all cells are used to make a demograph where the average fluorescence over the length axis of each cell is plotted, where all cells are ordered by cell length.

When 'timeD' is set to TRUE, bactKymo() makes kymographs of single cells. However, when 'percDiv' is set to true as well, bactKymo will make a kymograph of the average of all cell's fluorescence over their length axis, binning cells by the moment of division they are in (see `perc_Division()`).

Other settings which can be changes are the bin size (in how many groups is the average fluorescence intensity over the cell length calculated), whether the bins are placed such, that they represent the cell length at that moment (sizeAV=TRUE, takes more computing time), and when using a demograph, whether the extreme values are removed (cutoff_demograph).

The output will standardly be in the "viridis" color scale of the viridis package. However, this can be overwritten manually by adding a fill color scale to the plot.

To make computing time shorter while playing with the settings, it is possible to first prepare the kymographs using `prepForKymo()`. This is generally not necessary, but recommended when plotting single cell kymographs one by one.

Usage

```
bactKymo(originalCells,
          timeD = FALSE,
          dimension = "length",
          bins = 25,
          sizeAV = FALSE,
          cells = "all",
          prep = TRUE,
          percDiv = FALSE,
          cutoff_demograph = 0.975,
          mag,
          legend=TRUE)
```

Arguments

`originalCells` The output of extr_OriginalCells, which combines TIFF data with mesh data.

`timeD` if `timeD==TRUE`, the resulting kymographs will be sorted by time, not by cell length and will be made of single cells.

`dimension` default="length", but when put to "width" the kymographs will average over cell width.

`bins` the amount of bins the fluorescence intensity will be averaged over (per cell/time point)

`sizeAV` when `sizeAV==TRUE`, the kymographs will reflect the cell length (or width) on the y axes. Default to FALSE because the computing time is faster when there is no sizing.

`cells` default = "all". 'cells' is a numeric vector OR the word "all". When set to "all" ,all cells will be processed into a demograph (when timeD==FALSE) or a kymograph will be made of all cells (when timeD==TRUE). When a numeric vector of length 1 or more, the cells identified with the cell numbers in these vector will be processed into a demograph (when timeD == FALSE), or a kymograph will be made of all single cells (when timeD==TRUE). Finally, when only 1 cell number is identified, a kymograph will be made of this cell (when timeD == TRUE). It is not possible to make a plot of a single cell without time dimension.

`prep` default = TRUE. Inside the function, the data will be prepared so it is possible to plot kymographs. Sometimes, when you want to run `bactKymo()` multiple times on the same dataset, it can save time to prepare the dataset only once. For this you can use the function `prepForKymo()` first and use the as the argument `originalCells`, while setting 'prep' to FALSE.

`percDiv` this will call for `perc_Division()` and bin the average fluorescence over cell length (or width) in 10 groups, from just divided to almost dividing. It will, by default, plot all cells and only works with timelapse data.

`cutoff_demograph`
to make it easier to look at demographs of all cells, the highest 2.5 percent of the intensity values are removed from the demograph. set the cutoff to 1 to not remove anything, or put it lower to remove more values.

`mag` magnification converter (see `Pixels2um`. If used, the pixel values will be converted to micron.

`legend` when `legend==TRUE`, a legend will be shown on the right. when FALSE, the legend will be omitted.

Details

Many options are possible within `bactKymo()`. Make sure you set the time dimension & cells correctly! For examples, see the BactMAP wiki

Note that when you have an output of many plots, it is useful to use `gridExtra::grid.arrange()` to save all plots in a PDF, this makes the plots easier to investigate and saves loading time in between.

Value

One or multiple demographs/kymographs (see the details in `arguments`).

Author(s)

Renske van Raaphorst

Examples

```
#to plot a timelapse demograph of 1 cell:

kymo_cell4 <- bactKymo(originalCells = bactMAP::TurnedCell4, timeD=TRUE, sizeAV=TRUE)

#to change the color scheme:
OrangeHot <- getPalette("OrangeHot")
kymo_cell4 + scale_fill_gradient2(low=OrangeHot[1], mid=OrangeHot[2], high=OrangeHot[3])
```

bactMAP   *Analyze Bacterial Cell Segmentation And Fluoresence Data*

Description

Package to upload and uniformally analyze (bacterial) cell segmentation and fluorescence data, making it possible to combine output from different analysis tools into one or more datasets. Vizualize the data using ggplot2, statistically compare datasets and automatically produce a summary of the data.

Note

Compatible software outputs:

*MicrobeJ<www.microbej.com> *ObjectJ<https://sils.fnwi.uva.nl/bcb/objectj/> *ISBatch<http://singlemolecule.github.io/iSBatch/> *Oufti<www.oufti.org> *MicrobeTracker<www.microbetracker.org> *Morphometrics<https://simtk.org/projects/morphometrics> *SuperSegger<https://github.com/wiggins-lab/SuperSegger/wiki

Author(s)

Renske van Raaphorst<renske.vanraaphorst@unil.ch> University of Lausanne

References

van Raaphorst R*, Kjos M*, Veening JW (2017) Chromosome segregation drives division site selection in *Streptococcus pneumoniae. Proc Natl Acad Sci U S A.* 114(29):E5959-E5968.

combineDataframes *Function to combine data by condition (e.g. mutant vs wild type) and/or channel (e.g. GFP/RFP), meant for BactMAP imported data.*

Description

The function adds a column "condition" and/or "channel" to each dataframe in the given *listofdataframes*, after which it concatenates all dataframes. If not all columns in the dataframes match, it discards the columns which don't match. The function returns the concatenated dataframe "finalframe" and the original list of dataframes with the "condition" and/or "channel" column added to each dataframe.

Usage

```
combineDataframes(listofdataframes, listofconditions, listofchannels)
```

Arguments

`listofdataframes`
list of dataframes of which the common columns need to be combined

`listofconditions`
list of conditions corresponding to each dataframe. Needs to have the same order and length as listofdataframes.

`listofchannels`
list of channels corresponding to each dataframe. Needs to have the same order and length as listofdataframes.

Value

`finalframe` concatenated dataframe with columns "condition" and/or "channel" added

`originaldata` the input list of dataframes with columns "condition" and/or "channel" added

Author(s)

Renske van Raaphorst

Examples

```
GFPdata_WT <- extr.Oufti(file.choose())
RFPdata_WT <- extr.Oufti(file.choose())

RFPdata_mut  <- extr.Oufti(file.choose())
GFPdata_mut  <- extr.Oufti(file.choose())

combined <- combineDataframes(list(GFPdata_WT, RFPdata_WT, RFPdata_mut, GFPdata_mut),
                              listofconditions = list("WT", "WT", "mut", "mut"),
                              listofchannels=list("GFP", "RFP", "RFP", "GFP"))
```

createPlotList *Function to create descriptive plots of spot localization per group of cells*

Description

createPlotList() is a plotting function which combines detected fluorescent spot localization with mesh outline information to plot demographs, projections, histograms and a data summary in one go. These functions are all combined because they need similar data preparation. The idea is that after using this function, you have a good overview of the localization of your protein inside the cells. Sometimes this analysis can be enough to show where a protein localizes, but in other cases (when proteins are very mobile or localize heterogeneously, for instance), this may rather serve as a starting point for more detailed analysis. Therefore, the data generated to make these plots is also returned by the function.

Usage

```
createPlotList <- function(spotdata,
                           meshdata
                           groups=4,
                           colorpalette="GreenYellow",
                           mag="No_PixelCorrection",
                           AllPlot=T,
                           Xm="X",
                           Ym="Y",
                           viridis=FALSE,
                           showPlot=TRUE)
```

Arguments

spotdata	Data frame containing -at least- column with spot x-localizations, y-localizations and frame number. When the variable meshdata is not given, additional required columns are cell ID, relative x-localization ("l" or "Lmid") and relative y-localization ("d" or "Dum").
meshdata	Data frame containing -at least- the x and y coordinates of the cell meshes, the maximum cell width, maximum cell length, cell ID and frame. Optional.
groups	Number of groups the data will be divided in. Default = 4.
colorpalette	Color palette the heatmap plots will be plotted in. Default is "GreenYellow". Check which palettes are loaded by showColorPaletes() and add your own palette using addPallete()
mag	The name of the conversion factor for pixel to micrometer. Default is "No_PixelCorrection" which is 1. Check getPixels2um for the loaded conversionfactors and use addPixels2um("ConversionName", conversionfactor) to add a conversion factor to the list.
AllPlot	When TRUE, the output x/y plots will have sideplots with histograms of the spot localization. When FALSE, only the x/y plots will be plotted.
Xm and Ym	By default, the column names of the x/y coordinates in the MESH data frames are "X" and "Y". When this is not the case, one can tell the function by using Xm and Ym. Enter a character string with the name of the x-coordinate column for Xm, and the name of the y-coordinate column for Ym.

`viridis` when `viridis==TRUE`, the `colorpalette` needs to be one of the `viridis` color palettes ("A", "B", "C", "D", "E", "magma", "cividis", "viridis", "plasma" or "inferno")

`showPlot` when `showPlot==TRUE`, a menu will be shown after the function is finished, which enables the user to view the plot output of the function.

Value

A list consisting of the following items:

`$lengthplot` Heatmap histogram plot of the length-axis localizations of the spots ordered by the length of the cell.

`$widthplot` Heatmap histogram plot of the width-axis localizations of the spots ordered by the length of the cell.

`$qplots_separate`
If the outlines of the cell are not added to the function (dataframe `MESH` not indicated), the output will be a list of plots of length `inp`, with relative x/y coordinates of the spots plotted as a density function. The data is grouped by the variable `bycol`, which is `length` by default.If the outlines of the cells are added to the function (dataframe `MESH` indicated), the average cell outline of each group will be overlayed on top of each plot. If `AllPlot==TRUE`, histograms of the x and y coordinates of the spots will be added as sideplots.

`$qplots` for easy use, qplots is a `grob` arrangement of the group-plots `qplots_separate`: using `plot()` you can immediately view them in the Rstudio plots window.

`$plottotal` Relative x/y coordinates of all spots with average cell outline overlayed. If `AllPlot==TRUE`, the x/y coordinates will be plotted as histograms on the x and y axis of the plot.

`$histograms` 1D density plots of the spot localization on the length axis of the cells, grouped by `bycol` in the amount of groups defined by `groups`.

`$spotdata` data frame containing the groups, relative x/y localizations of the spots and cell length/width

`$meshdata` if `MESH` is included, the function returns a dataframe including the rotated meshes and groups.

`$data_summary` a character vector, or when `groups>1`, a list of character vectors summarizing the most important data features of each size group and the full dataset.

Note

This function is meant to give an overview of the data. Check the tutorials on the wiki page for examples of modifying individual plots.

Author(s)

Renske van Raaphorst

References

These plotting functions were developed for:

hrefvan Raaphorst, Renske, Morten Kjos, and Jan-Willem Veening. "Chromosome segregation drives division site selection in Streptococcus pneumoniae." Proceedings of the National Academy of Sciences 114.29 (2017): E5959-E5968.

Examples

```
##use data from example dataset

##1. detected peaks:
peaks <- bactMAP::DnaX_tracks


##2. mesh outlines:
mesh <- bactMAP::DnaX_mesh


##create list of plots, ordered by length, grouped in 6 groups.
GFPlist <- createPlotList(peaks, mesh, groups=6, colorpalette="GreenYellow", AllPlot=T)

#have a look at the "plottotal" plot
plot(GFPlist$plottotal)

#save PDF of the 6 grouped plots (qplots)
ggplot2::ggsave(GFPlist$qplots, filename="GroupPlots.PDF"))

#see summary:
GFPlist$data_summary
```

`DnaX_mesh` *DnaX and FtsZ timelapse of D39 S.pneumoniae. Cells are tracked using Oufti and converted to "mesh" dataframe using BactMAP::extr_Oufti*

Description

DnaX and FtsZ timelapse of D39 S.pneumoniae. Cells are tracked using Oufti and converted to "mesh" dataframe using BactMAP::extr_Oufti

Usage

```
DnaX_mesh
```

Format

A data frame with columns:

X x-coordinate of cell outline point (pixels)

Y y-coordinate of cell outline point (pixels)

cell cell identifier. unique per frame.

frame frame identifier. in this case, an image was taken every 20 seconds, so 1 frame corresponds to 20s.

num number indicating the order of cell outline points (useful for plotting polygons)

max.width maximum cell width in pixels

length the point of cell length from pole 0

steplength the distance (over the length axis) of this cell outline point from the previous cell outline point

max.length the maximum cell length of the given cell

xy indicates whether it’s the right or the left side of the cell (inherited from oufti)

area cell area in pixels(2)

angle the angle of the cells length axis to the horizontal line of the image.

Xmid the mid-point (pixel x-coordinate) of the cell

Ymid the mid-point (pixel y-coordinate) of the cell

X_rot the x-coordinate of the cell outline point (in pixels) when the cell length axis is horizontal, with mid-cell at (0,0)

Y_rot the y-coordinate of the cell outline point (in pixels) when the cell length axis is horizontal, with mid-cell at (0,0)

max_um the maximum cell length in micron

maxwum the maximum cell width in micron

Xrotum the x-coordinate of the cell outline point (in micron) when the cell length axis is horizontal, with mid-cell at (0,0)

Yrotum the y-coordinate of the cell outline point (in micron) when the cell lenth axis is horizontal, with mid-cell at (0,0)

Source

van Raaphorst, Renske, Morten Kjos, and Jan-Willem Veening. "Chromosome segregation drives division site selection in Streptococcus pneumoniae." Proceedings of the National Academy of Sciences 114.29 (2017): E5959-E5968.

Examples

```
## Not run:
 DnaX_mesh

## End(Not run)
```

DnaX_tracks *DnaX and FtsZ timelapse of D39 S.pneumoniae. DnaX spots are tracked using ISbatch and converted using extr_ISBatch*

Description

DnaX and FtsZ timelapse of D39 S.pneumoniae. DnaX spots are tracked using ISbatch and converted using extr_ISBatch

Usage

```
DnaX_tracks
```

Format

A data frame with columns:

x x-coordinate of detected DnaX spot

Y y-coordinate of detected DnaX spot

cell cell identifier. unique per frame.

frame frame identifier. in this case, an image was taken every 20 seconds, so 1 frame corresponds to 20s.

max.width maximum cell width in pixels

max.length the maximum cell length of the given cell

l relative localization of DnaX on the length axis of the cell.

d relative localization of DnaX on the width axis (diameter) of the cell

trajectory identifier of the DnaX track

trajectory_length the time (in frames) the trajectory could be followed

displacement_sq the displacement of the molecule (in pixels, as calculated by ISBatch)

Source

van Raaphorst, Renske, Morten Kjos, and Jan-Willem Veening. "Chromosome segregation drives division site selection in Streptococcus pneumoniae." Proceedings of the National Academy of Sciences 114.29 (2017): E5959-E5968.

Examples

```
## Not run:
 DnaX_tracks

## End(Not run)
```

`extr_ISBatch` *Upload ISBatch/Peakfitter data into R*

Usage

```
extr_ISBatch(dataloc)
```

Arguments

`dataloc` Path to the results table from the "peakfitter" option of the ISBatch imageJ plugin saved as .CSV or tab-delimited .TXT.

`seperator` When .CSV is not comma-separated but by something else, indicate this here.

Value

`cellList` copy of original data as dataframe

`spotframe` when data is made using ISBatch's peak fitter only, this data frame contains only x, y -coordinates and frame numbers. When trajectories are recorded, it also contains the squared-displacement, trajectory ID and trajectory length.

Author(s)

Renske van Raaphorst

References

http://singlemolecule.github.io/iSBatch/

Caldas, V.E., Punter, C.M., Ghodke, H., Robinson, A. and van Oijen, A.M., 2015. iSBatch: a batch-processing platform for data analysis and exploration of live-cell single-molecule microscopy images and other hierarchical datasets. *Molecular BioSystems*, 11(10), pp.2699-2708.

See Also

Use spotsInBox to combine the spot data with segmentation data from another program.

Examples

```
##Choose ISBatch peakfitter output file
dataloc <- file.choose()
##Convert to dataframe
spotframe <- extr_ISBatch(dataloc)
```

---

extr_Meshes *Generic function to upload cell outlines from a .csv or .txt file.*

Description

Output needs to have at least the following columns:

`X` - The x-coordinate of one cell outline point.

`Y` - The y-coordinate of one cell outline point.

`frame` - The image frame number.

`cell` - The cell identifyer (numeric).

Optional:

`max.length` - The cell length (in pixels)

`max.width` - The cell diameter (in pixels)

Any other parameters will be saved in the output dataframe cellList.

Usage

```
extr_Meshes(dataloc, sep = ",", turn = TRUE, mag)
```

Arguments

`dataloc`	The path to the .csv or .txt - file containing the mesh data.
`sep`	The separator of the columns in the .txt or .csv file, default = ","
`turn`	When `turn==TRUE`, the cells will be turned so that the mid-point is on coordinate [0,0] and the length axis parallel to the x axis.
`mag`	The magnification convertion factor (see addPixels2um). When given, the output will be also given in microns.

Value

`cellList`	the original input as dataframe
`mesh`	the mesh data compatible with BactMAP

Author(s)

Renske van Raaphorst

See Also

`pixel2um`

Examples

```
meshdata <- file.choose()

mesh_BM <- extr_Meshes(meshdata, sep=",", mag="100x_FRAP", turn=FALSE)
```

`extr_Morphometrics` *Upload cell outlines segmented using Morphometrics*

Description

The 'CONTOURS'-output from Morphometrics is uploaded and converted to a *mesh*-dataframe compatible with bactMAPs plotting functions. Find more examples of usage of `extr_Morphometrics` on the BactMAP `wiki`.

Usage

```
extr_Morphometrics(morphpath, mag, turncells = TRUE)
```

Arguments

`morphpath`	the path to the 'CONTOURS'- output of Morphometrics (for instance "mycells_01-01-2019_CONTOURS.mat").
`mag`	pixel conversion factor (see pixels2um). if added, the coordinates and cell length/width will be converted to micron
`turncells`	if set to TRUE (default), X and Y coordinates will be turned so that mid-cell is at [0,0], while the length axis of the cell's bounding box is parallel to the x axis.

Value

`cellList`	dataframe containing the original data extracted from Morphometrics.
`mesh`	dataframe containing the coordinates of the outlines of the cell, cell dimensions and if *turncells==TRUE*, the coordinates of each cell turned and repositioned to [0,0]
`pixel2um`	if *mag* was defined in the arguments, the conversion factor is saved in the output list.

Author(s)

Renske van Raaphorst

References

Ursell T, Lee TK, Shiomi D, et al. Rapid, precise quantification of bacterial cellular dimensions across a genomic-scale knockout library. BMC Biol. 2017;15(1):17. Published 2017 Feb 21. doi:10.1186/s12915-017-0348-8

Examples

```
#Upload Morphometrics output from bactMAP example files

Example_MM <-  file.choose()
Example_output <- extr_Morphometrics(Example_MM)

#check output:
summary(Example_output)

#plot cells
plotRaw(Example_output, frameN=1)
```

extr_ObjectJ *Upload and transform ObjectJ bounding box data.*

Description

ObjectJ detects a bounding box around each cell, which it uses to calculate spot localization, cell width and cell length along the medial axis. It is possible to save the bounding box points manually by defining them in the Columns section of the ObjectJ results. Here you can download an ObjectJ Settings file where the bounding box columns are already created.

Usage

```
extr_ObjectJ( dataloc,
              mag = "No_PixelCorrection",
         boundingBoxX = c("X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10", "X11"),
         boundingBoxY = c("Y1", "Y2", "Y3", "Y4", "Y5", "Y6", "Y7", "Y8", "Y9", "Y10", "Y11"),
              turn_meshes = TRUE)
```

Arguments

`dataloc` The path to the saved .CSV or .TXT file from ObjectJ.

`mag` Magnification conversion factor (see `pixels2um`)

`boundingBoxX` The column names of the 11 X-coordinates of the bounding box of the cells. Put these columns in your ObjectJ analysis manually or download the ObjectJ Settings here.

`boundingBoxY` The column names of the 11 Y-coordinates of the bounding box of the cells. Put these columns in your ObjectJ analysis manually or download the ObjectJ Settings here.

`turn_meshes` Default TRUE. When TRUE, the bounding boxes are all turned so the mid-point is at [0,0] while the length axis of the cell is parallel to the X-axis.

Details

ObjectJ can already convert it's data to micron for you! Therefore, the default setting for "mag" is "no_PixelCorrection". Check what your settings are before converting the data using BactMAP.

Value

`$cellList` The original CSV as dataframe.

`$mesh` Dataframe containing the X and Y coordinates of the bounding box and cell dimensions. When `turn_meshes=TRUE`, it also contains the coordinates of the turned meshes, cell angle (to the x axis of the image) and mid-point coordinates.

`GFPframe` Dataset contains self-defined (by the user) fluorescence information from the ObjectJ eg septal position or fluorescence intensity.

`$chainframe` Dataset contains information on cell chains: chain ID, length, width and if specified, fluorescence information.

Note

It is possible to define your own column names in ObjectJ - but (apart from the bounding box columns) bactMAP recognises the standard names only. See the [example objectJ settings file](https://github.com/vrrensk for the standard column names.

Author(s)

Renske van Raaphorst

References

https://sils.fnwi.uva.nl/bcb/objectj/

Vischer, Norbert OE, et al. "Cell age dependent concentration of Escherichia coli divisome proteins analyzed with ImageJ and ObjectJ." Frontiers in microbiology 6 (2015): 586.

Examples

```
 #Take ObjectJ file from R package environment

 #Using pipes:
 library(tidyverse)

ObjectJ_output <- system.file("extdata", "bsubtilis_box.txt", package="bactMAP") %>% extr_ObjectJ()

 #check summary
 ObjectJ_output %>% summary()

 #plot chain length distribution
 ggplot(ObjectJ_output$chainframe, aes(x=ChainAxis)) + geom_histogram()
```

extr_Original *Convert original TIFF image stack used to gather spot and cell outline data.*

Description

extr_OriginalStack() converts the original TIFF stack used to gather spot and cell outline data to a data frame containing the pixel x,y coordinates and the pixel value.

extr_OriginalCells() is connecting cell outlines to pixel values, using the output of extr_OriginalStack() and a mesh dataframe.

Usage

```
extr_OriginalStack(picloc)

extr_OriginalCells(imdatframe, mesh)
```

Arguments

	extr_OriginalStack:
	file path of the .TIFF image stack.
	extr_OriginalCells:
picloc imdatframe	output dataframe of extr_OriginalStack()
mesh	dataframe containing (at least) mesh coordinates, cell numbers and frame numbers.

Details

extr_OriginalStack needs the package "raster" to function.

extr_OriginalCells needs "SMDTools" and "raster" to function

Note

1. extr_OriginalCells is a heavy function since it's turning all pixels of each cell!

2. extr_OriginalStack works for single TIFFs and stacks, not hyperstacks. It only works with greyscale stacks.

Examples

```
#get tif
TIFFfilepath <- file.choose()

TIFFstack <- extr_OriginalStack(TIFFfilepath)

#plot stack
bactMAP::plotRaw(TIFFstack, frameN=1)

#get cell meshes
Meshpath <- file.choose()
meshes <- extr_Oufti(Meshpath)
```

```
#combine cells & tiff
extr_OriginalCells(TIFFstack, meshes$mesh)
```

extr_Oufti

Function to upload Oufti meshes and spot localizations directly from .Mat, .TXT or .CSV files to R.

Usage

```
extr_Oufti(matfile, mag="No_PixelCorrection", phylo=FALSE)
```

Arguments

`matfile` .MAT, .CSV or .TXT of oufti segmentation. Spot data, object data and fluorescence intensity will automatically be detected. Note: object data can only be extracted from .CSV or .TXT files for now!

`mag` The pixel-micron conversion factor. See bactMAP::getPixels2um() for more information. When not given, the factor will be 1.

`phylo` When TRUE, a list of child-parent nodes & phylo objects containing cell genealogy will be made. Takes very long, so default is set to FALSE

Value

A list containing:

`$cellList` The complete cellList as it is displayed in oufti's output.

`$mesh` A dataframe containing the following columns: `X, Y, cell, frame, num, max.width, length, st` See standard mesh output for more information.

When spot data is included:

`$spotframe` A dataframe containing the following columns: `l, d, x, y, position, adj_Rsquared, CI_xy, fr` See www.oufti.org for more information on the spot data output.

`$spots_relative`
A dataframe containing from the mesh data: max.width, max.length, area, cell, frame. from the spot data: l, d, x, y. Added: spot, totalspot, Lmid, Dum, max_um, maxwum, pole1, pole2.

`$pixel2um` the pixel to micron conversion factor used

When a CSV or TXT file containing object data:

`$objectframe` A dataframe containing the coordinates of detected objects, cell ID, frame ID and object ID: ob_x, ob_y, obnum, obpath, frame, cell, obID

`$object_relative`
A dataframe reminescent of spots_relative containing the following variables: "frame, cell, max.length, obnum, num, obpath, obID, max.width, Dum, Lmid, ob_out_x, ob_out_y, pole1, pole2

When "phylo" is set to TRUE:

`$timelapsedata` a list of: generation_dataframes: a dataframe with cells with common ancestor, generation_lists: phylo objects which can be used for tree plotting, spot_relative_list: list of spot localization dataframes of cells with common ancestor, and meshdata: a list of cell outline data of cells with common ancestor

Author(s)

Renske van Raaphorst

References

www.oufti.org

Paintdakhi A, et al. (2016) Oufti: an integrated software package for high-accuracy, high-throughput quantitative microscopy analysis. *Molecular Microbiology* 99(4):767-777.

Examples

```
##upload matlab oufti file
matfile <- file.choose()
##extract data using extr.Oufti()
output <- extr_Oufti(matfile)
##get spot dataframe
spots <- output$spotframe
##plot spots of frame no.1
ggplot2::ggplot(spots[spots$frame==1,], ggplot2::aes(x=x, y=y, color=as.factor(cell))) + ggplot2::geom_point
```

extr_Spots *Generic function for extraction of spot data*

Description

To upload spot localization data to R in the standard BactMAP format, save your spot data as a .txt or .csv file with (at least) the following columns:

`x` - the x coordinate of each spot

`y` - the y coordinate of each spot

`frame` - the number of the image frame

The function also recognises the following columns:

`trajectory` - ID variable for a spot trajectory

`trajectory_length` - the recorded length of each trajectory

`displacement_sq` - the squared displacement of each spot

Other columns will be saved in the output "cellList" (see below).

Usage

```
extr_Spots(dataloc, seperator = ",")
```

Arguments

`dataloc` The path to the .txt or .csv file containing the spot data.

`seperator` Indicates the column seperator of the .txt or .csv. Default = ","

Value

`cellList`	the original input data as dataframe
`spotframe`	dataframe containing the minimal required columns (x, y & frame) and if given, the extra columns (trajectory, trajectory_length & displacement_sq)

Author(s)

Renske van Raaphorst

Examples

```
spotpath <- file.choose()

spot_output <- extr_Spots(spotpath, seperator="/t")$spotframe
```

`extr_SuperSeggerCells` *Convert SuperSegger cell outlines to standard BactMAP format*

Description

Part of the *extr_*-functions, extr_SuperSeggerCells takes the mask created by SuperSegger of each cell (and if available, timepoint) and saves this as a *mesh* dataframe.

Usage

```
extr_SuperSeggerCells(loc, frames, mag, timelapse=FALSE, startframe=0)
```

Arguments

`loc`	The file path where SuperSegger's output files are located
`frames`	The number of xy frames (so not timepoints, just different locations!) imaged
`mag`	magnification conversion factor name (which is part of Pixels2um)
`timelapse`	set timelapse to TRUE if you are analyzing a timelapse movie.
`startframe`	default = 0. set to 1 if the first of your xy-locations is 1.

Details

Use *addPixels2um()* to add a new conversion factor.

Value

`cellList`	dataframe with content similar to SuperSegger's "Clist"
`mesh`	dataframe containing cell coordinates and dimensions
`pixel2um`	the magnification conversion factor used

Author(s)

Renske van Raaphorst

References

Stylianidou, Stella, et al. "SuperSegger: robust image segmentation, analysis and lineage tracking of bacterial cells." Molecular microbiology 102.4 (2016): 690-700.

`FtsZ_tracks` *DnaX and FtsZ timelapse of D39 S.pneumoniae. FtsZ spots are tracked using ISbatch and converted using extr_ISBatch*

Description

DnaX and FtsZ timelapse of D39 S.pneumoniae. FtsZ spots are tracked using ISbatch and converted using extr_ISBatch

Usage

```
FtsZ_tracks
```

Format

A data frame with columns:

x x-coordinate of detected dnaX spot

Y y-coordinate of detected dnaX spot

cell cell identifier. unique per frame.

frame frame identifier. in this case, an image was taken every 20 seconds, so 1 frame corresponds to 20s.

max.width maximum cell width in pixels

max.length the maximum cell length of the given cell

l relative localization of FtsZ on the length axis of the cell.

d relative localization of FtsZ on the width axis (diameter) of the cell

trajectory identifier of the FtsZ track

trajectory_length the time (in frames) the trajectory could be followed

displacement_sq the displacement of the molecule (in pixels, as calculated by ISBatch)

Source

van Raaphorst, Renske, Morten Kjos, and Jan-Willem Veening. "Chromosome segregation drives division site selection in Streptococcus pneumoniae." Proceedings of the National Academy of Sciences 114.29 (2017): E5959-E5968.

Examples

```
## Not run:
 FtsZ_tracks

## End(Not run)
```

`mesh` *Format of the Mesh dataframe*

Description

When extracting segmentation data, the main output is a dataframe called 'mesh'. The standard structure of this dataframe is shown below:

Format

A data frame with at least the following 7 variables:

- `X` a numeric vector indicating the X-coordinate of one mesh outline point.
- `Y` a numeric vector indicating the Y-coordinate of one mesh outline point.
- `cell` a numeric vector giving the cell ID (unique per image frame)
- `frame` a numeric vector giving the ID of the image frame
- `max.length` a numeric vector giving the cell length (in pixels)
- `max.width` a numeric vector giving the cell diameter (in pixels)
- `num` a numeric vector indicating the order of the cell outline points.

When the cells are turned (default) such, that the mid-point is at [0,0] and the length axis is parallel to the x axis, the following 5 variables are added:

- `angle` the angle of the cell length axis towards the 0 x-axis
- `Xmid` the X-coordinate of the cell's midpoint
- `Ymid` the Y-coordinate of the cell's midpoint
- `X_rot` the X-coordinate of the turned cell
- `Y_rot` the Y-coordinate of the turned cell

When a magnification conversion factor is indicated, the coordinates and lengths are calculated in micron. The following 4 variables are added:

- `Xrotum` the X-coordinate of the turned cell in micron
- `Yrotum` the Y-coordinate of the turned cell in micron
- `max_um` the cell length in micron
- `maxwum` the cell width in micron

In most cases, the cell area is also indicated:

- `area` The cell area in pixel^2

Finally, depending on the input programs, some other variables can be added to the output. See the original programs output documentation for clarification.

`micron` *Get back the unicode character for the micro-sign + m (micron).*

Description

Use micron() in a character string, for instance when changin the axis labels of a plot, to get the unicode for the micro-sign + m.

Usage

```
micron()
```

Value

character string: 'u00b5m'

Examples

```
paste("This is the sign for micron:", bactMAP::micron(), sep=" ")
```

`object` *Object DataFrames*

Description

A dataframe containing object information, either `objectframe` or `object_relative`.

Usage

```
These dataframes can be used in various of the bactMAP functions: \code{\LinkA{plotTracks}{plotTrack
```

Format

In the case of `objectframe`, the most basic of the two, the object coordinates and cell information are given:

`ob_x` the x-coordinate (in pixels) of the object shape outline

`ob_y` the y-coordinate (in pixels) of the object shape outline

`obnum` the number of the object inside one cell

`obpath` number indicates the order of the object coordinates, so a polygon will be drawn correctly

`frame` the number of the image frame

`cell` the number of the bacterial cell

`obID` unique object identifier

In the case of `object_relative`, the following colums are added:

`max.length` the cell length of the given cell the object is in.

num a rank for the cell size - the smaller num, the smaller the cell size

max.width the cell width of the given cell this object is in.

Dum the relative localization of the mid-point of the object on the width axis of the cell, in micron.

Lmid the relative localization of the mid-point of the object on the length axis of the cell, in micron.

ob_out_x the relative x-coordinate (in micron) of the object shape outline to the cell

ob_out_y the relative y-coordinate (in micron) of the object shape outline to the cell

pole1 the distance of one cell pole to mid-cell

pole2 the distance of the other cell pole to mid-cell. Since mid-cell is placed at [0,0], this is always the negative of pole1.

max_um max.length in micron

maxwum max.width in micron

Details

An example of an object dataframe can be taken from the bactMAP package. How is shown in "examples" below.

Examples

```
##Take Oufti output from bactMAP package:

VanFL <- system.file("extdata", "WT_obj.txt", package = "bactMAP")

##extract data using extr_Oufti

ex_VanFL <- extr_Oufti(VanFL, mag="100x_DVMolgen")

##View objectframe:
View(ex_VanFL$objectframe)

##And object_relative
View(ex_VanFL$object_relative)
```

objectInBox *Connect detected objects to detected cell meshes; turn all objects & cells in the same orientation for relative comparison.*

Description

ObjectInBox is a function based on spotsInBox - it returns a dataframe which has the same variables as the "object_relative" dataframe.

Usage

```
objectInBox(objectdata, meshdata, mag = "No_PixelCorrection")
```

Arguments

objectdata	Object dataframe with at least the variables: "ob_x" & "ob_y" (object outline coordinates), "frame" and "obID" (object identifier).
meshdata	Mesh dataframe.
mag	Magnification converter (see pixels2um).

Value

An object_relative dataframe (see `object`).

Author(s)

Renske van Raaphorst

Examples

```
#example data:

WTmesh <- bactMAP::vanFL_meshWT

WTobjects <- bactMAP::vanFL_objWT

#these objects are already connected to cells. but no harm in re-doing:

WTbox <- objectInBox(WTobjects, WTmesh, mag="100x_DVMolgen")
```

`perc_Division` *Filter timelapse data by growth speed and whether cells underwent a whole division cycle.*

Description

The function takes timelapse dataframes including cell size, frame and cell ID and return a dataframe with information on growth speed, whether the cell underwent a full division (and how many divisions), and a binning of each cell division in percentages. Next to this, it gives a summary of the parameters in the input dataframe per binned division point. It also plots the individual cell growth over binned division, as well as the average cell growth.

Usage

```
perc_Division(timelapse, av = TRUE, plotgrowth = TRUE)
```

Arguments

`timelapse`	A dataframe from timelapse data containing at least the variables `frame`, `cell` and `max.length`.
`av`	When `av==TRUE`, the function returns a dataframe with the variables of the input dataframe averaged per binned division moment ("percentage_binned").
`plotgrowth`	When `plotgrowth==TRUE`, the function returns a plot of the cell growth by binned division point, growth speed and nth division.

Details

This function estimates whether a division is a "full" division by a set of cutoffs. In some cases, the segmentation data gives this information already. When this is the case, it can be more straightforward to use this data instead of this function.

Value

1. A dataframe `timelapse` containing the following variables on top of the variables already in the input dataframe:

`division` the nth division in the timelapse this cell is making

`min_frame` the frame number where this division starts

`max_frame` the frame where the cell is dividing

`av_length` the mean cell length over the whole division

`length_var` the variance of the cell length during the whole division

`division_time` the time this division takes (in frames)

`fulldivision` indicates whether the division is a full division (1 = full)

`coeff` the coefficient of the linear growth function of the cell for this division - indicator of growth speed

`growth` the divisions are grouped in 4 groups: no growth ("none"), fast, medium and slow growth.

`percentage` if the moment of cell birth is 0 and cell division is 100, the percentage gives how far the cell is from dividing at this moment

`percentage_binned` percentage grouped in 10 groups, for averaging

2. A dataframe `mean_by_percentage`, where the variables are averaged over `percentage_binned`.
3. A ggplot `plot_growth` showing the cell size over the percentage of division.
4. A ggplot `plot_avgrowth` showing the average cell size over the percentage of division.

Author(s)

Renske van Raaphorst

Examples

```
#this example won't have much meaning because the timelapse was quite short.
#but analysis is quick because the dataset is not very big:
#for a better example, download our DnaX dataset from veeninglab.com/bactmap

percD <- perc_Division(bactMAP::FtsZ_tracks)

percD$plot_avgrowth
```

`Pixels2um` *Add and look up magnification factor "pixels2um"*

Description

Functions to check the pixel to micron conversion factors loaded into bactMAP and to add new conversion factors. The new conversion factors will be saved throughout the running R session.

Usage

```
getPixels2um()

addPixels2um(pixelName, pixels2um)
```

Arguments

`pixelName`	A character string identifying your new conversion factor
`pixels2um`	A numerical value to convert the pixel values to microns

Value

getPixels2um() and addPixels2um("ConversionName", conversionfactor) both return the currently loaded conversion factors as a list of vectors, where the first item of each vector is the conversionfactorname and the second item is the numeric conversion factor.

Note

If you don't know your pixel conversion factor, open your original images in FIJI/ImageJ and check the Properties of the image.

Author(s)

Renske van Raaphorst

Examples

```
## To check the currently loaded pixel conversion factors:
getPixels2um()

## To add your pixel conversion factor to bactMAP:
addPixels2um("MyPixel2um", 0.543)
```

`plotCellsTime` *Function to plot cell fluorescence from a TIFF over time.*

Usage

```
plotCellsTime(celdat, updown = T, movie = F, viridisoption = "magma", cellN, minf, maxf)
```

Arguments

`celdat`	TIFF and mesh outline data from a timelapse movie as it was obtained by running `extr_OriginalCells`.
`updown`	When updown==`TRUE`, a horizontal cell tower is made, otherwise, the cells are displayed left-right. `movie`==TRUE will overwrite this and return an animation instead.
`movie`	When `movie` == `TRUE`, an animation made with `gganimate` is returned.

viridisoption	The color palette used. The default palettes are part of the viridis color palettes (default = "magma").
cellN	The cell number or numbers to be displayed. When not indicated, a list of plots of all cells will be made.
minf	The minimum frame to be displayed.
maxf	The maximum frame to be displayed

Value

A plot, list of plots or animation of a single cell's fluorescence over time.

Author(s)

Renske van Raaphorst

Examples

```
##In BactMAP, the raw data of one cell is saved as "TurnedCell4":

#to plot a left-right tower:
plotCellsTime(bactMAP::TurnedCell4, updown=F, viridisoption="inferno")

#or to create an animation:
plotCellsTime(bactMAP::TurnedCell4, movie=T)
```

plotObjects *Create plots similar to createPlotList using intracellular object outlines instead of points.*

Description

Plot projection of fluorescent objects inside cells in one or more cell groups based on cell length.

Usage

```
plotObjects(obdat,
            meshdat,
            groups = 1,
            cellcolor = "black",
            objectcolor = "green",
            transparency = 0.1,
            getdata = FALSE)
```

Arguments

obdat	Object dataset "object_relative" as it is extracted using one of the "extr_"-functions.
meshdat	A mesh dataset as it is extracted using one of the "extr_"-functions (optional).
groups	The amount of groups to devide the cell projections in (by cell length). Default = 1.

`cellcolor` The color of the cells in the cell projections. Default = black.

`objectcolor` The color of the objects in the projections. Default = green.

`transparency` The transparency of the objects and cells in the overlay. Default = 0.1. When analyzing ~1000 cells, it is advised to set this number to ~0.02 or less.

`getdata` When getdata==TRUE, the function returns a dataset with the grouped cells next to the plot. Default is set to FALSE.

Value

A ggplot of cell and object projections showing the relative localization of the intracellular objects, grouped by cell length.

Author(s)

Renske van Raaphorst

See Also

`createPlotList`

Examples

```
#get objects & meshes from WT D39 cells stained with Van-FL, analyzed with oufti.

library(bactMAP)

VanFL_mesh <- bactMAP::VanFL_meshWT
VanFL_obj <- bactMAP::VanFL_objWT

#plot objects

plotObjects(VanFL_obj, VanFL_mesh, groups=4, transparency=0.1)

#change colors/layout

library(ggdark)
library(ggplot2)
plotObjects(VanFL_obj, VanFL_mesh, groups=4, transparency=0.1, cellcolor="grey40", objectcolor="magenta") + g
```

`plotOverlay` *Display spot- and/or object-localization of different color channels and/or experimental conditions in one plot.*

Usage

```
plotOverlay(meshdata,
            spotdata,
            objectdata,
            by = "both",
            type = "all",
            quantiles = 1,
            quantiles_by = "max.length",
            mag,
            objectcolor = c("#E69F00", "#56B4E9", "#009E73", "#F0E442"),
            spotcolor = c("#0072B2", "#D55E00", "#CC79A7", "000000"),
            histogram_outline = NA,
            his_scales = "free")
```

Arguments

`meshdata` A `mesh` dataframe generated by combining dataframes with the function `combineDataframes`. Optional.

`spotdata` A `spot` dataframe generated by combining dataframes with the function `combineDataframes`. Optional.

`objectdata` A `object` dataframe generated by combining dataframes with the function `combineDataframes`. Optional.

`by` Either "both", "channel" or "condition", indicating which variables are to be plotted.

`type` The type of output plot. Options are: "histogram", "length", "width", "projection" or "all". Examples of each plot are given below.

`quantiles and quantiles_by`
When "projection" or "all" is chosen, the cells can be split into groups by `quantiles_by`: one of the other columns existing in each dataset (default = `max.length`). Default for `quantiles` = `1`, so no grouping.

`mag` magnification convertion factor, needed when cell values are not yet converted to microns. See `getPixels2um`.

`objectcolor` Indicates the color(s) of the objects (if included). Default colorset is a range of 4 colors ("#E69F00", "#56B4E9", "#009E73" and "#F0E442").

`spotcolor` Indicates the color(s) of the spot data (if included). Default colorset is a range of 4 colors ("#0072B2", "#D55E00", "#CC79A7" and "000000").

`histogram_outline`
By default, the histograms displayed have no outlines. Indicate a hex value or a colorname to change this.

`his_scales` Inherited from ggplot2::geom_facet_grid(): when cells are grouped into different groups, by default, the histograms y-axis is scaled free per group. Setting `his_scales` to "fixed" will make the scale the same for each group.

Value

A plot or a list of plots:

`histogram` Histogram plot of the localization distribution, if `quantiles` are set, grouped by `by_quantiles`.

`length` Dotplot showing the localization of the spots and/or objects on the cell's length axis on the y-axis, and the cells ordered by cell length on the x-axis.

`width` Dotplot showing the localization of the spots and/or objects on the cell's width axis on the y-axis, and the cells ordered by cell length on the x-axis.

`projection` Cell projections showing the relative object localization, spot localization and cell shape; if `quantiles` are set, grouped by `by_quantiles`.

Examples

```
##get data from BactMAP package
#condition one: wild type
WT_objects <- VanFL_objWT
WT_mesh <- VanFL_meshWT

#condition 2: dmapZ
DM_objects <- VanFL_objDM
DM_mesh <- VanFL_meshDM

comboMesh <- combineDataframes(list(WT_mesh, DM_mesh), listofconditions=list("WildType", "DeltaMapZ"))
comboObjects <- combineDataframes(list(WT_objects, DM_objects), listofconditions=list("WildType", "DeltaMapZ

plotOverlay(meshdata=comboMesh$finalframe,
            objectdata=comboObjects$finalframe,
            by="condition",
            quantiles=4,
            type="projection")
```

---

plotRaw — *Plot TIFF, cell contours and/or spot localization on the original x/y space.*

Description

To check segmentation, view TIFF images and/or show the localization of spots inside cells. ggplot2-based, so other geoms and theme layers can be added.

Usage

```
plotRaw(tiffdata, meshdata, spotdata, frameN = 1, xrange, yrange, viridisoption = "inferno", meshco
```

Arguments

`tiffdata` (list of) dataframe(s) containing *x* and *y* coordinates of pixels and their *values*. Can be generated by loading a TIFF into R using bactMAP::extr_OriginalStack()

`meshdata` dataframe containing (at least) *X* and *Y* coordinates of cells outlines, the *frame* number of the image and *cell* numbers.

`spotdata` dataframe containing (at least) *x* and *y* coordinates of spots and the *frame* number of the image.

`frameN` number of the frame to be displayed. default = 1.

`xrange` vector with minimum and maximum x coordinates to be displayed (by default all is shown).

`yrange` vector with minimum and maximum y coordinates to be displayed (by default all is shown).

`viridisoption` one of the viridis color scale options (default = "inferno") for the displayed TIFF. options: "magma", "viridis", "cividis", "plasma" and "inferno"

`meshcolor` color of the cell outlines. default = "white"

`spotcolor` color for displayed spots. default = "yellow"

`valuerange` vector with minumum and maximum pixel values to show TIFF, use this option to increase contrast

Value

A plot showing either a TIFF, mesh outlines, spot coordinates or a combination of the three.

Note

Renske van Raaphorst

See Also

extr_Original, viridis

Examples

```
#load TIF image from BactMAP examples. go to veeninglab.com/bactmap to download.

TIF_image <- file.choose()

#use extr_-function to turn into list of dataframes

TIF_df <- bactMAP::extr_OriginalStack(TIF_image)

#plot image
plotRaw(TIF_image)

#zoom in and add title
plotRaw(TIF_image, xrange=(c(200,500)), yrange=c(400,600)) + ggplot2::ggtitle("This is a TIF image")
```

plotTracks *Plot tracked spots and/or objects over time in 1 or 2 dimensions.*

Usage

```
plotTracks( meshdata,
            spotdata,
            objectdata,
            tracks = TRUE,
            ignore_singles = FALSE,
            movie = "none",
            timepalette_lines = "viridis",
            timepalette_fill = "magma",
            dimension = c("length", "width"),
            turn_cells = TRUE,
            mag,
            cell = "all",
            transparency = 0.2
            )
```

Arguments

meshdata
: A mesh dataframe containing cell outlines.

spotdata
: A Spot Data dataframe containing spot localizations.

objectdata
: A object dataframe containing fluorescent object outlines.

tracks
: When tracks == TRUE, the track of individual spots over time will be shown. When tracks == FALSE, only the spots will be shown.

ignore_singles
: When ignore_singles == FALSE, all detected spots will be shown. When ignore_singles ==TRUE, only those spots will be shown which are tracked over time.

movie
: When movie=="none" (default), the output will be a static plot. When movie == "gganimate", gganimate (version => 1.0.3) will be used to make a GIF animation. Finally, when movie = "plotly", plotly will be used to make an interactive HTML-based animation.

timepalette_lines
: A viridis palette for both the spots & tracks as the outlines of the meshes, where color indicates the time in frames. Default = "viridis".

timepalette_fill
: A viridis palette for the fill of the objects, where color indicates the time in frames. Default = "magma".

dimension
: Either "length", "width", or c("length", "width")(default). Indicates the dimensions which will be plotted next to time (frames).

turn_cells
: When cell projections are plotted (both cell length & width), it is possible to plot cells with their original x/y coordinates, or turn them so their length axis is parallel to the x-axis with mid-cell at 0. When plotting only one cell dimension, this command is ignored and cells are turned by default.

mag
: The magnification converter (see pixels2um).

cell
: The cell number or numbers as a numeric vector (e.g. c(1,2,10)), or (default) "all". When more then one cell are indicated, the function returns a plot faceted by cell number.

transparency
: Transparency of both the objects as the cell polygons. The cell polygons are set to be 5 times more transparent than the object polygons.

Value

A plot or a faceted plot of the trajectory of the cell shape, fluorescent spots and/or the localization of detected fluorescent objects in the cell over time. The plots can either by location on the length-axis over time, location on the width-axis over time, or x/y location and shape over time.

Author(s)

Renske van Raaphorst

Examples

```
##Get example data from bactMAP package

##tracked FtsZ
FtsZ <- bactMAP::FtsZ_tracks
```

```
 ##and corresponding meshes
 meshes <- bactMAP::DnaX_mesh

 #plot ftsZ on length axis/time
plotTracks(meshdata=meshes, spotdata=FtsZ_tracks, dimension="length", mag="100x_DVMolgen", tracks=TRUE, cel
```

plotTreeBasic *Plot cell genealogy information as networks using ggtree.*

Description

This function is using the ggtree package to show cell genealogy combined with cell information, e.g. fluorescence intensity. This part of bactMAP is under development - plotting cell trees is possible, but is not (yet) as straight-forward. Check the tutorials on veeninglab.com/bactmap to get more information on how to use this function.

Usage

```
plotTreeBasic(phylo,
              extradata,
              yscalechange = FALSE,
              showClade = FALSE,
              layout = "rectangular",
              ydata,
              cellNumber,
              open.angle,
              linesize = 1,
              linecolor = "black",
              lines = TRUE,
              colors = FALSE)
```

Arguments

`phylo` The `$generation_lists` output of `extr_SuperSegger` or `extr_Oufti`.

`extradata` Data connected to the `$generation_lists`: `$generation_dataframes`. When this is added, the output of the plot looks the same, but it is possible to add layers to the plots with cell information using the `ggtree`-commands. See the tutorials for more information.

`yscalechange` Default = `FALSE`. When `TRUE`, an y-axis is added to the tree. Only works when also indicating what needs to be plotted on the y-axis (indicate by putting the column name of choice from `extradata` as input in `ydata`).

`showClade` When `showClade==TRUE`, one clade will be highlighted. Decide which one by putting a cell number in `cellNumber`.

`layout` Inherited from `ggtree`; this decides the shape of the plotted tree. Options: `"rectangular", "slanted", "fan", "circular", "radial", "unrooted", "equal_angle", "`

`ydata` Indicate the column name of the variable you want to be plotted on the y-axis of the tree (when `yscalechange==TRUE`).

`cellNumber` Indicate the number of the cell of which you want to highlight the clade (when `showClade==TRUE`).

`open.angle` When `layout=="fan"`, indicate the size of the open angle (in degrees) here.

`linesize` Thickness of the lines in the tree.

`linecolor` Indicate the color of the lines in the tree.

`lines` When highlighting a clade, set `lines` to `TRUE` when you want to do that by linetype.

`colors` When highlighting a clade, set `colors` to `TRUE` when you want to do that by color.

Value

A `ggtree` object which can be edited using the `ggtree` commands.

Author(s)

Renske van Raaphorst

References

The ggtree cookbook by Guangchuang Yu.

Examples

```
#this code requires other packages:
requires(ggtree)
requires(ggplot2)

#load example data from bactMAP package
myTreeData <- bactMAP::ssbB_phylo
myFluoData <- bactMAP::ssbB_meanfluo

#plot Tree with attributes
plotTreeBasic(myTreeData$phylos, myTreeData$data_attributes) +
 geom_point(aes(x=x-branch.length, color=fluormean), size=6) + #same as before, but different dot size
  geom_tippoint(aes(color=fluormean_D), size=6) + #added tip points, different dot size
 geom_label(aes(x=x-branch.length, label=label), color=NA, fill="white", alpha=0.5, size=1) +
   #I also make a seethrough background label so you can see the text over the colored dots
  geom_text(aes(x=x-branch.length, label=label), size=3) +
  scale_color_viridis_c(option="magma") + #add different color scale, for fun
  theme(legend.position="right") + #add legend
  coord_flip()
```

prepForKymo *Prepare whole dataset for single cell kymographs.*

Description

Extension from `bactKymo`, this function prepares a dataset for kymograph plotting. Recommended if you plan to plot a lot of kymographs from one dataset one by one; this will speed up the process. If you don't plan to do this, use `bactKymo` immediately.

Usage

```
prepForKymo(turnedCells, dimension = "length", bins = 25, sizeAV = FALSE)
```

Arguments

`turnedCells`	Output from `extr_OriginalCells`, which includes the pixel values (of the fluorescence signal) in each cell.
`dimension`	Default = "length", "width" is also possible. This decides on which dimension the kymograph will be calculated.
`bins`	Default = 25. The amount of bins in which the average fluorescence over length or width will be calculated.
`sizeAV`	When `sizeAV == TRUE`, the bins are placed along the y axis of the plot according to cell size. This takes more computation time.

Value

A dataframe suitable for `bactKymo()`.

`Spot Data` *Spot Data: dataframes describing fluorescent spot coordinates and (optional) trajectories*

Description

When fluorescent spots are detected, the output "spotframe" (and when mesh data is included "spots_relative") is created. This dataset contains the x and y coordinates of the spots, the image frame in which the spots occurs, and if cell information is given, the relative localization of the cells. When spots where tracked over time, the spot trajectory, squared displacement and track length are given.

Format

spotframe contains:

`x` a numeric vector giving the x coordinate of the spot on the image frame

`y` a numeric vector giving the y coordinate of teh spot on the image frame

`frame` a numeric vector giving the number of the image frame

when spots where tracked, the following columns are added:

`trajectory` a numeric vector giving an index for the spot trajectory. when spot is not part of a trajectory, this vector == -1

`displacement_sq` a numeric vector giving the spots squared displacement (in pixels^2)

`trajectory_length` a numeric vector giving the length (in frames) of the spot trajectory

when cell information is giving, the spot data is listed in "spots_relative", which include the above and the following:

`l or Lmid` a numeric vector giving the relative localization over the length axis of the cell. l is given in pixels, Lmid in micron `d or Dum` a numeric vector giving the relative localization over the width axis of the cell. `d` is giving in pixels, Dum in micron

`cell` the index of the cell the spot belongs to

`max.width` a numeric vector giving the length of the cell

`max.length` a numeric vector giving the width of the cell

`pip` when pip == 1, the spot is found inside a cell

`spot` a numeric vector indexing the spots per cell

`totalspot` the total amount of spots per cell

`cellframe` optional: the cell index and frame index pasted together with a "." as separator, to give a unique index to each cell

---

`spotsInBox` *Put spots in boxes*

Description

Takes a dataframe containing Mesh x/y coordinates and a dataframe containing Spots/Object x/y coordinates and returns the relative localization of the Spots/Objects inside the meshes, and the relative coordinates of the meshes from the mid-point of each mesh, where the longest axis of the cell is horizontal in the x/y plane.

Usage

```
spotsInBox(spotdata, meshdata, Xs = "x", Ys = "y", Xm = "X", Ym = "Y")
```

Arguments

`spotfile` dataframe containing spot coordinates. Needs at least the columns `$frame, $x` and `$y`.Possible to define x/y column names in the function input arguments `Xs` and `Ys`.

`MESH` dataframe containing mesh coordinates. Needs at least the columns `$frame, $cell, $x` and `$y`. Possible to define x/y column names in the function input arguments `Xm` and `Ym`.

`Xs` column in dataframe `spotfile` containing the x-coordinates of the spots or object points. Default = "x".

`Ys` column in dataframe `spotfile` containing the y-coordinates of the spots or object points. Default = "y".

`Xm` column in dataframe `MESH` containing the x-coordinates of the mesh outline points. Default = "X".

`Ym` column in dataframe `MESH` containing the y-coordinates of the mesh outline points. Default = "Y".

Value

A list of two data frames:

`spots_relative:`

`$x`	x-coordinate of the spot or object point (same as original dataframe)
`$y`	y-coordinate of the spot or object point (same as original dataframe)
`$cell`	identifies the cell the spot/object point belongs to
`$frame`	frame number of original image stack
`$max.width`	maximum cell width
`$length`	length of the cell
`$L`	relative location of the spot/object point to the length axis of the cell
`$D`	relative location of the spot/object point to the width axis of the cell

`mesh:`

`$X`	x-coordinate of the mesh contour point (same as original dataframe)
`$Y`	y-coordinate of the mesh contour point (same as original dataframe)
`$cell`	identifies the cell
`$frame`	frame number of the original image stack
`$max.width`	maximum cell width
`$length`	length of the cell
`$X_rot`	x-coordinate of the mesh point when the cell is turned with the length axis horizontally, midpoint at coordinate (0,0)
`$Y_rot`	y-coordinate of the mesh point when the cell is turned with the length axis horizontally, midpoint at coordinate (0,0)

Author(s)

Renske van Raaphorst

See Also

`shotGroups`

`pnt.in.poly`

Examples

```
##get spot data from ISBatch
spots <- bactMAP::extr.ISBatch(file.choose())$spotframe

##get mesh data from Oufti
mesh <- bactMAP::extr.Oufti(file.choose())$mesh

##get relative spot and mesh locations
outlist <- spotsInBox(spots, mesh, Xm="x", Ym="y")
```

`ssbB_meanfluo` *Dataset retrieved from movie previously published in Moreno-Gamez et al., 2017 Rhrefhttps://www.ncbi.nlm.nih.gov/pubmed/29021534pubMED. Competence is induced with CSP and ssbB-GFP is expressed. The timeing between cells differs slightly. Cells are tracked using SuperSegger.*

Description

Dataset retrieved from movie previously published in Moreno-Gamez et al., 2017 pubMED. Competence is induced with CSP and ssbB-GFP is expressed. The timeing between cells differs slightly. Cells are tracked using SuperSegger.

Usage

```
ssbB_meanfluo
```

Format

A data frame with columns:

node the number of the node in the genealogy tree

birth birthframe of this cell

cell cell identifier. unique per frame.

death death frame (or frame the cell divides) of this cell

edgelength the time (in frames) of the cell division

fluorsum the sum of the total fluorescence inside the cell. inherited from SuperSegger (see `the SuperSegger Wiki` for more information)

fluormean the mean of the total fluorescence inside the cell. inherited from SuperSegger (see `the SuperSegger Wiki` for more information)

fluorsum_D the sum of the fluorescence at cell division. inherited from SuperSegger (see `the SuperSegger Wiki` for more information)

fluormean_D the mean of the total fluorescence at cell division. inherited from SuperSegger (see `the SuperSegger Wiki` for more information)

parent the node of the parent of this cell

child1 the cell number of the daughter cell of this cell

child2 the cell number of the other daughter cell of this cell

root the root of the genealogy tree. by default, this is 0.

nodelabel the cell number corresponding to the node

Source

Moreno-GC!mez, Stefany, et al. "Quorum sensing integrates environmental cues, cell density and cell history to control bacterial competence." Nature communications 8.1 (2017): 854.

Examples

```
## Not run:
 ssbB_meanfluo

## End(Not run)
```

`ssbB_network` *Dataset retrieved from movie previously published in Moreno-Gamez et al., 2017 Rhrefhttps://www.ncbi.nlm.nih.gov/pubmed/29021534pubMED. Competence is induced with CSP and ssbB-GFP is expressed. The timeing between cells differs slightly. Cells are tracked using SuperSegger.*

Description

Dataset retrieved from movie previously published in Moreno-Gamez et al., 2017 pubMED. Competence is induced with CSP and ssbB-GFP is expressed. The timeing between cells differs slightly. Cells are tracked using SuperSegger.

Usage

```
ssbB_network
```

Format

A phylo object containing the genealogy information of this timelapse movie.

iGRAPH network dataset containing the same information as `ssbB_phylos`

Source

Moreno-GC!mez, Stefany, et al. "Quorum sensing integrates environmental cues, cell density and cell history to control bacterial competence." Nature communications 8.1 (2017): 854.

Examples

```
## Not run:
 plot(ssbB_network)

## End(Not run)
```

`ssbB_phylos` *Dataset retrieved from movie previously published in Moreno-Gamez et al., 2017 Rhrefhttps://www.ncbi.nlm.nih.gov/pubmed/29021534pubMED. Competence is induced with CSP and ssbB-GFP is expressed. The timeing between cells differs slightly. Cells are tracked using SuperSegger.*

Description

Dataset retrieved from movie previously published in Moreno-Gamez et al., 2017 pubMED. Competence is induced with CSP and ssbB-GFP is expressed. The timeing between cells differs slightly. Cells are tracked using SuperSegger.

Usage

```
ssbB_phylos
```

Format

A phylo object containing the genealogy information of this timelapse movie.

Phylogenetic tree with 103 tips and 95 internal nodes.

Source

Moreno-GC!mez, Stefany, et al. "Quorum sensing integrates environmental cues, cell density and cell history to control bacterial competence." Nature communications 8.1 (2017): 854.

Examples

```
## Not run:
 ssbB_phylos

## End(Not run)
```

`TurnedCell4` *DnaX and FtsZ timelapse of D39 S.pneumoniae. This dataframe contains the fluorescence and mesh information of the FtsZ channel of cell number 4. The data is retreived by uploading the TIFF of FtsZ-RFP using extr_OriginalStacks and subsequently running extr_OriginalCells with this and the dataset bactMAP::DnaX_mesh. For disk space reasons, only cell 4 is saved here. Get the whole dataset on veeninglab.com/bactmap.*

Description

DnaX and FtsZ timelapse of D39 S.pneumoniae. This dataframe contains the fluorescence and mesh information of the FtsZ channel of cell number 4. The data is retreived by uploading the TIFF of FtsZ-RFP using extr_OriginalStacks and subsequently running extr_OriginalCells with this and the dataset bactMAP::DnaX_mesh. For disk space reasons, only cell 4 is saved here. Get the whole dataset on veeninglab.com/bactmap.

Usage

```
TurnedCell4
```

Format

A data frame with columns:

x x-coordinate of the pixel

y y-coordinate of the pixel

cell cell identifier. unique per frame.

frame frame identifier. in this case, an image was taken every 20 seconds, so 1 frame corresponds to 20s.

max.width maximum cell width in pixels

max.length the maximum cell length of the given cell

pointN number of specific pixel. unique per cell and frame

values intensity of the pixel

X_rot pixel x coordinate when the cell is turned so that the length axis is horizontal and mid-cell is at 0,0.

Y_rot pixel y coordinate when the cell is turned so that the length axis is horizontal and mid-cell is at 0,0.

xt the point marking one corner of the square pixel (x-coordinate)

yt the point marking one corner of the square pixel (y-coordinate)

area the cell area (pixel squared)

Source

van Raaphorst, Renske, Morten Kjos, and Jan-Willem Veening. "Chromosome segregation drives division site selection in Streptococcus pneumoniae." Proceedings of the National Academy of Sciences 114.29 (2017): E5959-E5968.

Examples

```
## Not run:
 TurnedCell4

## End(Not run)
```

`VanFL_meshDM` *VanFL dataset: cell outlines (mesh) DM*

Description

Cells were stained with fluorescent vancomycin and imaged as described in van Raaphorst, Kjos & Veening, 2017. The cells and fluorescent objects where segmented using Oufti and imported in R using BactMAP's extr_Oufti function.

Usage

```
VanFL_meshDM
```

Format

a "mesh" data frame (mesh) describing the shape and localization of fluorescent vancomycin in S. pneumoniae cells where the gene mapZ was replaced by a chloramphenicol resistance marker (DmapZ::cmR).

Source

van Raaphorst, Renske, Morten Kjos, and Jan-Willem Veening. "Chromosome segregation drives division site selection in Streptococcus pneumoniae." Proceedings of the National Academy of Sciences 114.29 (2017): E5959-E5968

Examples

```
## Not run:
 View(VanFL_objWT)

## End(Not run)
```

VanFL_meshWT *VanFL dataset: cell outlines (mesh) WT*

Description

Cells were stained with fluorescent vancomycin and imaged as described in van Raaphorst, Kjos & Veening, 2017. The cells and fluorescent objects where segmented using Oufti and imported in R using BactMAP's extr_Oufti function.

Usage

```
VanFL_meshWT
```

Format

a "mesh" data frame (mesh) describing the shape and size of Wild Type S. pneumoniae cells.

Source

van Raaphorst, Renske, Morten Kjos, and Jan-Willem Veening. "Chromosome segregation drives division site selection in Streptococcus pneumoniae." Proceedings of the National Academy of Sciences 114.29 (2017): E5959-E5968

Examples

```
## Not run:
 View(VanFL_meshWT)

## End(Not run)
```

VanFL_objDM *VanFL dataset: objects mapZ mutant*

Description

Cells were stained with fluorescent vancomycin and imaged as described in van Raaphorst, Kjos & Veening, 2017. The cells and fluorescent objects where segmented using Oufti and imported in R using BactMAP's extr_Oufti function.

Usage

```
VanFL_objDM
```

Format

an "object_relative" data frame (`object`) describing the shape and localization of fluorescent vancomycin in S. pneumoniae cells where the gene `mapZ` was replaced by a chloramphenicol resistance marker (`DmapZ::cmR`).

Source

van Raaphorst, Renske, Morten Kjos, and Jan-Willem Veening. "Chromosome segregation drives division site selection in Streptococcus pneumoniae." Proceedings of the National Academy of Sciences 114.29 (2017): E5959-E5968

Examples

```
## Not run:
View(VanFL_objDM)

## End(Not run)
```

VanFL_objWT *VanFL dataset: objects WT*

Description

VanFL dataset: objects WT

Usage

```
VanFL_objWT
```

Format

an "object_relative" data frame (`object`) describing the shape and localization of fluorescent vancomycin in Wild Type S. pneumoniae cells.

Cells were stained with fluorescent vancomycin and imaged as described in van Raaphorst, Kjos & Veening, 2017. The cells and fluorescent objects where segmented using Oufti and imported in R using BactMAP's extr_Oufti function.

Source

van Raaphorst, Renske, Morten Kjos, and Jan-Willem Veening. "Chromosome segregation drives division site selection in Streptococcus pneumoniae." Proceedings of the National Academy of Sciences 114.29 (2017): E5959-E5968

Examples

```
## Not run:
View(VanFL_objWT)

## End(Not run)
```

Index