

# Package ‘qtlc’

January 4, 2016

**Type** Package

**Title** Densitometric Analysis of Thin-Layer Chromatography Plates

**Version** 1.0

**Date** 2016-01-03

**Author** Ivan D. Pavicevic

**Maintainer** Ivan D. Pavicevic <ivanp84@gmail.com>

**Description** Densitometric evaluation of the photo-archived quantitative thin-layer chromatography (TLC) plates.

**Depends** tiff, rgl, plot3D, graphics, grDevices, stats, utils

**License** GPL-2

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-01-04 17:46:53

## R topics documented:

areadens2D . . . . .	2
areadensnoisecompare2D . . . . .	3
createTLC . . . . .	4
matrices2D . . . . .	5
matrixtoimage . . . . .	6
noisepoly2D . . . . .	7
normalize2D . . . . .	8
picmatrixTIFF . . . . .	8
print.qtlc . . . . .	9
print.summary.qtlc . . . . .	10
Rf . . . . .	11
rotatev . . . . .	12
s3D . . . . .	12
select2D . . . . .	13
showRf . . . . .	14

showtlc2D . . . . .	15
showtlc2D.qtlc . . . . .	15
showtlc3D . . . . .	16
spot2D . . . . .	17
summary.qtlc . . . . .	18
summat2D . . . . .	19

## Index 21

---

areadens2D	<i>Area vs. x-axis</i>
------------	------------------------

---

### Description

Function charts density plot of a single spot following x-axis.

### Usage

```
areadens2D(object, spot = NULL, plot = TRUE, returndf = TRUE,
           ptype = "o", ...)
```

### Arguments

object	S3 object of working TLC
spot	Number of the spot (counted left to right).
plot	Boolean, TRUE default and displays densitometric distribution.
returndf	Boolean, TRUE by default, returns data . frame with x and Area values.
ptype	Point type for the plot. Default "o" (Uses same values as type variable from plot function)
...	Additional parameters (for plot type function).

### Value

Returns data . frame with x and Area values.

### Author(s)

Ivan D. Pavicevic, <ivanp84@gmail.com>

### Examples

```
# A test example
fname01 <- system.file("extdata", "test025to100sp.tiff", package="qtlc")
testTLC <- createTLC(fname01, RGB=FALSE)
print(testTLC)

# now we'll imitate interactive spot2D function,
```

```

# and create spots coordinates automatically,
# for interactive version run testTLC <- spot2D(testTLC)
testTLC$spots$x <- c(40.93354, 83.18687, 121.59899, 160.01111, 203.54485,
                    239.39616, 280.36909, 320.06161, 362.31494, 399.44666,
                    439.13919, 480.11211, 518.52423, 559.49716, 599.18969)
testTLC$spots$y <- c(198.3160, 198.3160, 199.2833, 198.3160, 198.3160,
                    198.3160, 198.3160, 198.3160, 197.3487, 198.3160,
                    199.2833, 198.3160, 199.2833, 199.2833, 199.2833)

testTLC <- select2D(testTLC, 30, 30)
testTLC <- matrices2D(testTLC)
testTLC <- summat2D(testTLC)

# and now test the areadens2D for each spot
par(mfrow=c(3,3))
for(i in 1:15) {
  areadens2D(testTLC, spot=i, ptype="l")
}

```

---

```
areadensnoisecompare2D
```

*Shows areadens2D plus background noise as segmented line*

---

## Description

Plots area-densities of the spot and background.

## Usage

```
areadensnoisecompare2D(object, spot = NULL, plot = TRUE, returndf = TRUE,
  ptype = "o", ...)
```

## Arguments

object	S3 object of working TLC
spot	Number of the spot (counted left to right).
plot	Boolean, TRUE default and displays densitometric distribution.
returndf	Boolean, TRUE by default, returns data.frame with x and Area values.
ptype	Point type for the plot. Default "o" (Uses same values as type variable from plot function)
...	Additional parameters (for plot type function).

## Value

Returns data.frame with x and Area values.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:
#see areadens2D first
areadensnoisecompare2D(object, spot=3, ptype="1")

## End(Not run)
```

---

createTLC	<i>Creates TLC S3 object</i>
-----------	------------------------------

---

**Description**

Create matrix from TLC image

**Usage**

```
createTLC(ttiff, turnv = TRUE, ...)
```

**Arguments**

ttiff	File name of the TIFF image with scanned TLC plate.
turnv	Boolean value determines to turn vertically data in the matrix. TRUE generates turned image which is useful for Cartesian coordinates, because without turning the coordinate system begins in the left corner of the monithor and rises left and down.
...	Additional parameters for TIFF image manipulation.

**Value**

An object of class `qt.lc`, that contains TLC matrix and descriptions. The object contains:

file_name	File name of of the TIFF image from which the TLC matrix was created.
mat	TLC matrix (or matrices if intensities Red, Green and Blue channels are not combined.)
spots	Coordinates of marked spots (using function <code>spot2D</code> ).

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

## Examples

```
#Creates test matrix.
# RGB channels stay separated, or
# intensities are combined.
fname01 <- system.file("extdata", "testTIFF.tiff", package="qtlc")
testTLC <- createTLC(fname01, RGB=TRUE)
print(testTLC)
```

---

matrices2D	<i>Creates spots matrices</i>
------------	-------------------------------

---

## Description

Using spots locations and areas this function creates a matrix for each spot.

## Usage

```
matrices2D(object, ...)
```

## Arguments

object	S3 object of working TLC
...	Additional graphical parameters. (At this time just experimental)

## Value

Returns S3 object with new variable `object$spot_matrices` which is a three dimensional matrix (width, height, and pixel intensity values).

## Author(s)

Ivan D. Pavicevic, <ivanp84@gmail.com>

## Examples

```
## Not run:
## just makes spot matrices for selected spots of the object
object <- matrices2D(object)

## End(Not run)
```

---

matrixtoimage	<i>Converts matrix back to image</i>
---------------	--------------------------------------

---

### Description

Using S3 qtlc object, extracts the matrix and converts to image plot.

### Usage

```
matrixtoimage(object, show = TRUE, bkg = "thistle", axes = FALSE,  
              xlab = "", ylab = "", ...)
```

### Arguments

object	S3 object of working TLC.
show	Boolean, default TRUE. Shows the plot of the image.
bkg	If show is TRUE, then defines background color. Default is "thistle".
axes	Boolean, default FALSE. Shows x,y axes if TRUE.
xlab	Label of the x-axis.
ylab	Label of the y-axis.
...	Additional graphical parameters.

### Value

Returns image as matrix suitable for plot, or other graphics functions.

### Author(s)

Ivan D. Pavicevic, <ivanp84@gmail.com>

### Examples

```
# Converts test image to matrix,  
# then matrix back to image.  
fname01 <- system.file("extdata", "testTIFF.tiff", package="qtlc")  
testTLC <- createTLC(fname01, RGB=FALSE)  
print(testTLC)  
matrixtoimage(testTLC, bkg="white")
```

---

`noisepoly2D`*Polynomial estimation of the image noise.*

---

**Description**

Estimation, and noise removal using polynomial model.

**Usage**

```
noisepoly2D(object, gd = 20, power = 5, col = "green")
```

**Arguments**

<code>object</code>	S3 object of the working TLC
<code>gd</code>	Defines position of the center of the rectangular samples of the image background (above or below located spots).
<code>power</code>	Order of the polynome.
<code>col</code>	Color of the borders of the rectangles for bkg samples.

**Value**

Returns S3 object with new variables.

<code>object\$noise_mat</code>	The 3D matrix (width, height, number of spots)
<code>object\$noise_fit</code>	Linear model for the polynomial fit
<code>object\$noise_fit_spot_sums</code>	Sums of the noise samples areas

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
#object is a tlc with 14 spots, and selection 50x80(wxh)  
object <- noisepoly2D(object, gd=80, power=13)  
  
## End(Not run)
```

normalize2D

*Normalization of the matrix values*

---

**Description**

Experimental function. Normalize matrix data.

**Usage**

```
normalize2D(mat)
```

**Arguments**

mat                    Matrix of the TLC plate.

**Value**

Normalized matrix.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
new_mat <- normalize2D(mat)  
  
## End(Not run)
```

---

picmatrixTIFF*Converting TIFF to matrix.*

---

**Description**

Mostly internal function used by createTLC function. Additional parameters from createTLC goes there (RGB, comb).

**Usage**

```
picmatrixTIFF(ff, RGB = TRUE, comb = c(0.3, 0.59, 0.11))
```

**Arguments**

ff	TIFF file
RGB	Boolean, TRUE - keeps Red, Green and Blue intensities as three matrices. FALSE - using comb to combine intensities.
comb	Vector, combines intensities according to luma. A vector containing three values for R, G and B conversion.

**Value**

Returns combined intensities matrix, or separated R, G, B matrices.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
#Internal function used by createTLC(...)  
  
## End(Not run)
```

---

print.qtlc                      *Print Method for qtlc object*

---

**Description**

Redefined print method.

**Usage**

```
## S3 method for class 'qtlc'  
print(x, ...)
```

**Arguments**

x	S3 object of the working TLC.
...	Additional parameters for the print method

**Value**

Prints qtlc S3 object details

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
print(object)  
  
## End(Not run)
```

---

print.summary.qtlc      *Summary method for qtlc S3 object*

---

**Description**

Summary method for qtlc S3 object

**Usage**

```
## S3 method for class 'summary.qtlc'  
print(x, ...)
```

**Arguments**

x	S3 object of the working TLC.
...	Additional parameters.

**Value**

Summary.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
summary(object)  
  
## End(Not run)
```

---

Rf	<i>Retention factor (Rf)</i>
----	------------------------------

---

**Description**

Calculates Rf values of the spots based on the marked start and stop of the solvent path.

**Usage**

```
Rf(object, sf = F)
```

**Arguments**

object	S3 object of the working TLC
sf	Boolean, default FALSE - Start and Front should be marked. If TRUE, Start and Front were defined.

**Value**

Returns S3 object with new variables.

object\$Rf_start	Location of the solvent start on the TLC plate
object\$Rf_front	Location of the solvent end on the TLC plate
object\$Rf	Rf values of the spots

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
#for more interactive variant; locate using mouse  
object <- Rf(object, sf=TRUE)  
  
## End(Not run)
```

---

rotatev	<i>Rotate vertically</i>
---------	--------------------------

---

**Description**

Rotate entire matrix vertically. Mostly internal function.

**Usage**

```
rotatev(mat)
```

**Arguments**

mat            The matrix.

**Value**

Rotated matrix.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
new_mat <- rotatev(mat)  
  
## End(Not run)
```

---

s3D	<i>Internal function used by showtlc3D</i>
-----	--

---

**Description**

Internal function used by showtlc3D

**Usage**

```
s3D(mat, ogl, grey, ...)
```

**Arguments**

mat	Matrix with x,y,Intensity dimensions
ogl	If TLC 3D plot use OpenGL library for fast and interactive 3D plot. (This functionality is based on the rgl package.) Otherwise the function is based on the plot3D package.
grey	Boolean, if TRUE, then tlc is greyscaled. Default value is FALSE.
...	Additional graphics parameters.

**Value**

None.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:
#Internal function.

## End(Not run)
```

---

select2D	<i>Selects spots areas</i>
----------	----------------------------

---

**Description**

Based on the located spots centers (manually with mouse and function spot2D) this function defines spots areas.

**Usage**

```
select2D(object, w, h, col = "white")
```

**Arguments**

object	S3 object of the working TLC.
w	Width of the spot area.
h	Height of the spot area.
col	Color of the border (default white)

**Value**

Return S3 object with new variable object\$mat\_cell which is list with "w" and "h" values.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
select2D(object, 80, 50)  
  
## End(Not run)
```

---

showRf

*Shows Rf on the plot*

---

**Description**

Shows prior analysed Rf on the new plot of the 2D matrix.

**Usage**

```
showRf(object, col = "green", adjust = NULL, cex = 0.6)
```

**Arguments**

object	S3 object of the working TLC
col	Color of the lines.
adjust	Adjustment for the space of the text. Default value is usually just OK.
cex	A zoom factor for the text.

**Value**

None.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
showRf(object)  
  
## End(Not run)
```

---

showtlc2D	<i>Show TLC matrix as 2D plot</i>
-----------	-----------------------------------

---

**Description**

Using TLC matrix width, height, and intensity parameters this function plot 2D heatmap of the TLC matrix.

**Usage**

```
showtlc2D(object, ...)
```

**Arguments**

object	S3 object of the working TLC
...	Additional parameters

**Value**

None

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
showtlc2D(object)  
  
## End(Not run)
```

---

showtlc2D.qtlc	<i>Show TLC matrix as 2D plot</i>
----------------	-----------------------------------

---

**Description**

Using TLC matrix width, height, and intensity parameters this function plot 2D heatmap of the TLC matrix.

**Usage**

```
## S3 method for class 'qtlc'  
showtlc2D(object, specific = NULL, RGB = "", main = "",  
           correction = TRUE, grey = FALSE, ...)
```

**Arguments**

object	S3 object of the working TLC
specific	Matrix of the specific spot (from object\$spot_matrices)
RGB	RGB matrices (if they are present in the object) are separated on the plot. Values of the RGB = "R", or "G", or "B".
main	Main title of the plot.
correction	Experimental option, currently not in use.
grey	Boolean, if TRUE, then TLC plate is greyscaled. Default value is FALSE.
...	Additional graphical parameters

**Value**

None

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:
showtlc2D(object, specific=object$spot_matrices[3], grey=TRUE)

## End(Not run)
```

---

showtlc3D	<i>Shows 3D plot of the TLC matrix.</i>
-----------	---

---

**Description**

The function uses TLC matrix width, height and intensity values to make 3D plot.

**Usage**

```
showtlc3D(object, spot = NULL, ogl = FALSE, RGB = NULL, grey = FALSE,
...)
```

**Arguments**

object	S3 object of the working TLC.
spot	If the specific spot should be represented in 3D, but not entire TLC matrix. (Spot number is given as value, and spots are counted from left to right.)
ogl	If TLC 3D plot use OpenGL library for fast and interactive 3D plot. (This functionality is based on the rgl package.) Otherwise the function is based on the plot3D package.

RGB	If RGB matrices are present in the object, choose between R, G, or B.
grey	Boolean, if TRUE, then tlc is greyscaled. Default value is FALSE.
...	Additional graphics parameters.

**Value**

None.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
# Tests 3D plot of the entire matrix
fname01 <- system.file("extdata", "test025to100sp.tiff", package="qtlc")
testTLC <- createTLC(fname01, RGB=FALSE)

# now we'll imitate interactive spot2D function,
# and create spots coordinates automatically,
# for interactive version run testTLC <- spot2D(testTLC)
testTLC$spots$x <- c(40.93354, 83.18687, 121.59899, 160.01111, 203.54485,
                    239.39616, 280.36909, 320.06161, 362.31494, 399.44666,
                    439.13919, 480.11211, 518.52423, 559.49716, 599.18969)
testTLC$spots$y <- c(198.3160, 198.3160, 199.2833, 198.3160, 198.3160,
                    198.3160, 198.3160, 198.3160, 197.3487, 198.3160,
                    199.2833, 198.3160, 199.2833, 199.2833, 199.2833)

testTLC <- select2D(testTLC, 30, 30)
testTLC <- matrices2D(testTLC)
testTLC <- summat2D(testTLC)

# 3D without OpenGL, shows only spot 13
showtlc3D(testTLC, spot=13, ogl=FALSE, grey=FALSE)
# without openGL and greyscaled
showtlc3D(testTLC, spot=13, ogl=FALSE, grey=TRUE)
#openGL showtime
showtlc3D(testTLC, spot=13, ogl=TRUE)
```

---

spot2D

*Locate spots manually.*

---

**Description**

The function should be used after 2D TLC matrix was plotted. After function call, the user should manually locate centers of the spots using mouse. (Left click for locate, right for the end of the process.)

**Usage**

```
spot2D(object, col = "white", ...)
```

**Arguments**

object	S3 object of the working TLC.
col	Color of the spot locator (default is white)
...	Additional parameters.

**Value**

S3 object with 'object\$spots' added.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
print(object)  
object <- spot2D(object)  
  
## End(Not run)
```

---

summary.qtlc

*Summary method for qtlc S3 object*

---

**Description**

Summary method for qtlc S3 object

**Usage**

```
## S3 method for class 'qtlc'  
summary(object, ...)
```

**Arguments**

object	S3 object of the working TLC.
...	Additional parameters for the summary method

**Value**

Summary.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
## Not run:  
summary(object)  
  
## End(Not run)
```

---

summat2D

*Summarize matrices*

---

**Description**

The function summarize matrices areas of the located spot matrices.

**Usage**

```
summat2D(object)
```

**Arguments**

object            S3 object of working TLC

**Value**

Returns S3 object with new values object\$spot\_sums.

**Author(s)**

Ivan D. Pavicevic, <ivanp84@gmail.com>

**Examples**

```
# This interactive example shows the most  
# common usage of the qtlc library.  
fname01 <- system.file("extdata", "test025to100sp.tiff", package="qtlc")  
testTLC <- createTLC(fname01, RGB=FALSE)  
print(testTLC)  
  
# now using mouse select the spots with testTLC <- spot2D(testTLC)  
# but, for automatic tests, we'll imitate that step...  
testTLC$spots$x <- c(40.93354, 83.18687, 121.59899, 160.01111, 203.54485,  
                  239.39616, 280.36909, 320.06161, 362.31494, 399.44666,  
                  439.13919, 480.11211, 518.52423, 559.49716, 599.18969)  
testTLC$spots$y <- c(198.3160, 198.3160, 199.2833, 198.3160, 198.3160,  
                  198.3160, 198.3160, 198.3160, 197.3487, 198.3160,
```

```
199.2833, 198.3160, 199.2833, 199.2833, 199.2833)

# and now the select2D selects 30x30 pixels areas around spots
testTLC <- select2D(testTLC, 30, 30)

# forming spots matrices
testTLC <- matrices2D(testTLC)

# and finally summarizing spots areas
testTLC <- summat2D(testTLC)

#eventually we'll examine the linear model
C <- rep(c(0.25, 1, 6.25, 25, 100), each=3) #imaginative concentrations
#now creates data frame with values
testTLC.df <- data.frame(C, testTLC$spot_sums)
names(testTLC.df) <- c("Concentration", "Signal")
# now the linear model
testTLC.lm <- with(testTLC.df, lm(Signal ~ Concentration))
# and finally the plot
plot(testTLC.df)
abline(testTLC.lm)
summary(testTLC.lm)
```

# Index

\*Topic **array**

createTLC, 4

\*Topic **device**

createTLC, 4

areadens2D, 2

areadensnoisecompare2D, 3

createTLC, 4

matrices2D, 5

matrixtoimage, 6

noisepoly2D, 7

normalize2D, 8

picmatrixTIFF, 8

print.qtlc, 9

print.summary.qtlc, 10

Rf, 11

rotatev, 12

s3D, 12

select2D, 13

showRf, 14

showtlc2D, 15

showtlc2D.qtlc, 15

showtlc3D, 16

spot2D, 17

summary.qtlc, 18

summat2D, 19