HowTo_FlimDiagRam

Julien Godet

22/04/2020

How to ?

This is a short vignette showing how to use the *flimDiagRam* package.

Contents

Installation

To be able to use the functions and utilities of the *flimDiagRam package*, you need first to install the package. This can be done by installing the last updated version directly from github.

```
if(!require(devtools)){install.packages("devtools")}
if(!require(flimDiagRam)){devtools::install_github("jgodet/flimDiagRam")}
```

Then, call the package

```
library(flimDiagRam)
```

‘ **Note:** installation needs to be done only once. On the contrary, packages must be called on every new R sessions

For this vignette, we will also install two additional packages

```
if(!require(utilitR)){devtools::install_github("jgodet/utilitR")}
if(!require('tidyverse')){install.packages('tidyverse')}
```

Read the *.asc files

The first step is to read the data. For this vignette, we can use file examples available on github. These files correspond to data of a singly labelled protein (labelled with eGFP) acquired at two different days of experiment.

```
path <- "https://raw.githubusercontent.com/jgodet/flimDiagRam/master/data/"
tauFilePath <- paste(path, "PAS406_001_t1.asc", sep="")
photFilePath <- paste(path, "PAS406_001_photons.asc", sep="")

data <- getData(pathTau = tauFilePath, pathPhoton = photFilePath, label = "Image1")
```

Let’s check everything went OK

```
head(data)
```

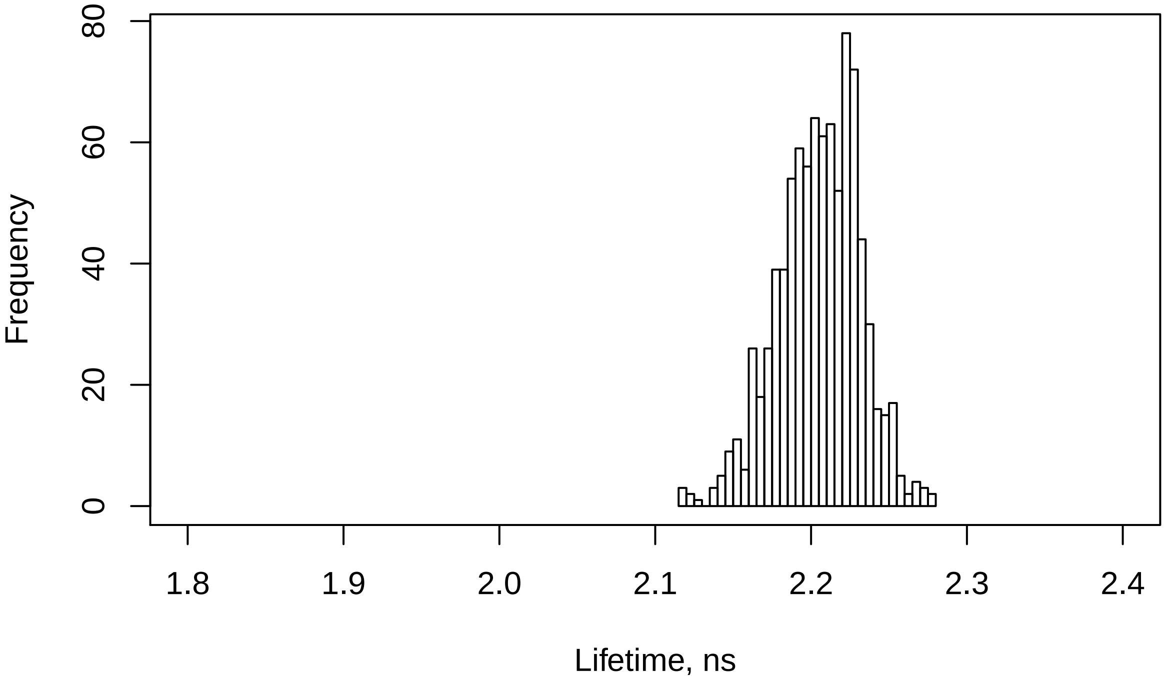
```
##       tau1d photons row col    ind
## 579 2144.20     762  67   5 Image1
## 580 2139.48    2377  68   5 Image1
## 581 2138.70    1537  69   5 Image1
## 582 2119.50     156  70   5 Image1
## 706 2153.60     272  66   6 Image1
## 707 2166.15    2415  67   6 Image1
```

This table (data.frame) gives for each line the lifetime value (tau1d in ps), the intensity (photons), and the coordinates (row and column index) of a pixel of the FLIM image.

Basic data exploration

Histogram of the lifetime distribution

```
hist(data$tau1d/1000, main="", xlab="Lifetime, ns", xlim=c(1.8, 2.4), breaks = 50)
box()
```

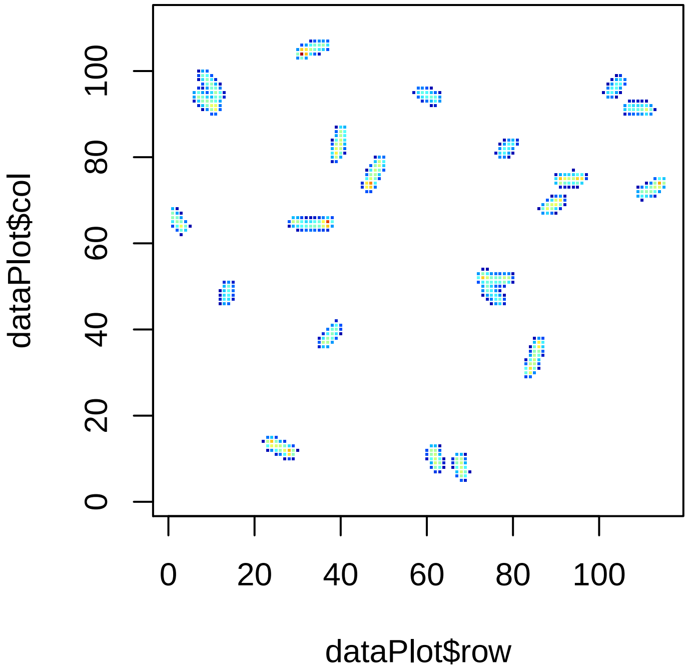


Aggregate data by bacteria

Plot the image

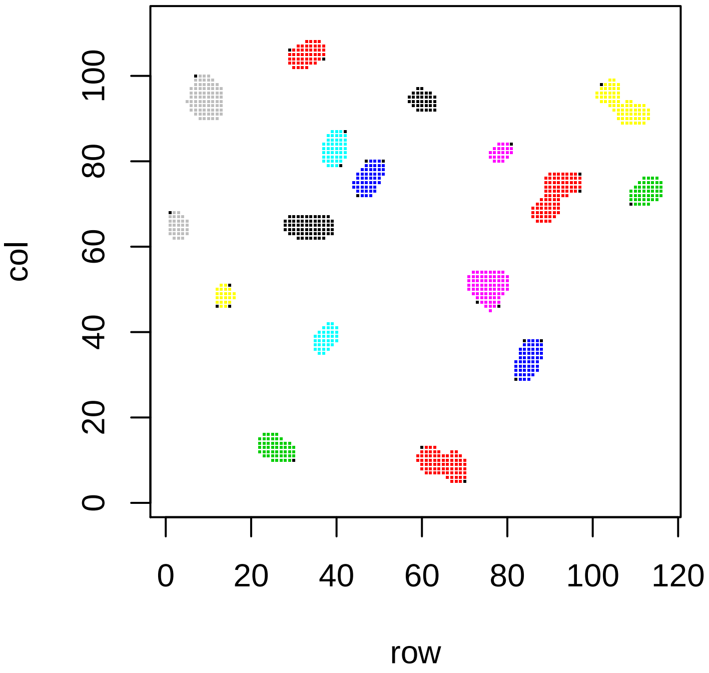
```
dataPlot <- data %>% dplyr::filter(photons>1000) %>%
  dplyr::select(row, col, photons)
plot(x = dataPlot$row, y = dataPlot$col ,pch='.' , col=utilitR::contColor(dataPlot$photons),asp=1)
```

```
## Loading required package: colorRamps
```



We will use the *dbscan* package to cluster pixels belonging to the same bacteria and *tidyverse* package to explore the data

```
if(!require('dbscan')){install.packages('dbscan')}
require(dbscan)

res <- dbscan(data[,c("row","col")], eps = 1, minPts = 5)
data$bacteria <- res$cluster
plot(data[,c("row","col")], col = res$cluster + 1L, pch='.', asp=1)
```



Note : The cluster labelled “0” gathers pixel that were not attributed to any cluster.

Note 2 : Some bacteria are not correctly segmented using this basic clustering approach. It might be a good option to filter clusters according to their area (or number of pixels in the cluster) to discard the one corresponding to multiple bacteria

Now we need to aggregate data by cluster

```
Im1_sumup <- data %>% filter(bacteria != 0) %>%
  group_by(bacteria) %>%
  dplyr::summarise( nPixels =n(),
             tau_mean = mean(tau1d, na.rm = T),
             tau_sd = sd(tau1d, na.rm = T),
             tau_med = median(tau1d, na.rm = T),
             label = unique(ind)) %>%
  filter(nPixels<80)

Im1_sumup
```

```
## # A tibble: 16 x 6
##    bacteria nPixels tau_mean tau_sd tau_med label
##       <int>   <int>    <dbl>  <dbl>   <dbl> <chr>
##  1        1      73    2209.   24.9   2214. Image1
##  2        2      49    2222.   14.3   2223. Image1
##  3        3      52    2207.   20.4   2213. Image1
##  4        4      34    2173.   17.7   2178. Image1
##  5        5      71    2181.  21.6    2181. Image1
##  6        6      22    2207.  23.2    2213. Image1
##  7        7      29    2202.  15.0    2208. Image1
##  8        8      63    2207.  17.6    2207. Image1
##  9       10      44    2213.  18.0    2216. Image1
## 10       11      48    2222.  27.1    2224. Image1
## 11       12      45    2222.  18.4    2224. Image1
## 12       13      22    2178.   8.07   2180. Image1
## 13       14      70    2189.  19.2    2194. Image1
## 14       15      75    2215.  20.5    2218. Image1
## 15       16      31    2187.  12.0    2187. Image1
## 16       17      47    2231.  30.2    2230. Image1
```

We have some nice descriptive statistics of 16 individual bacteria.

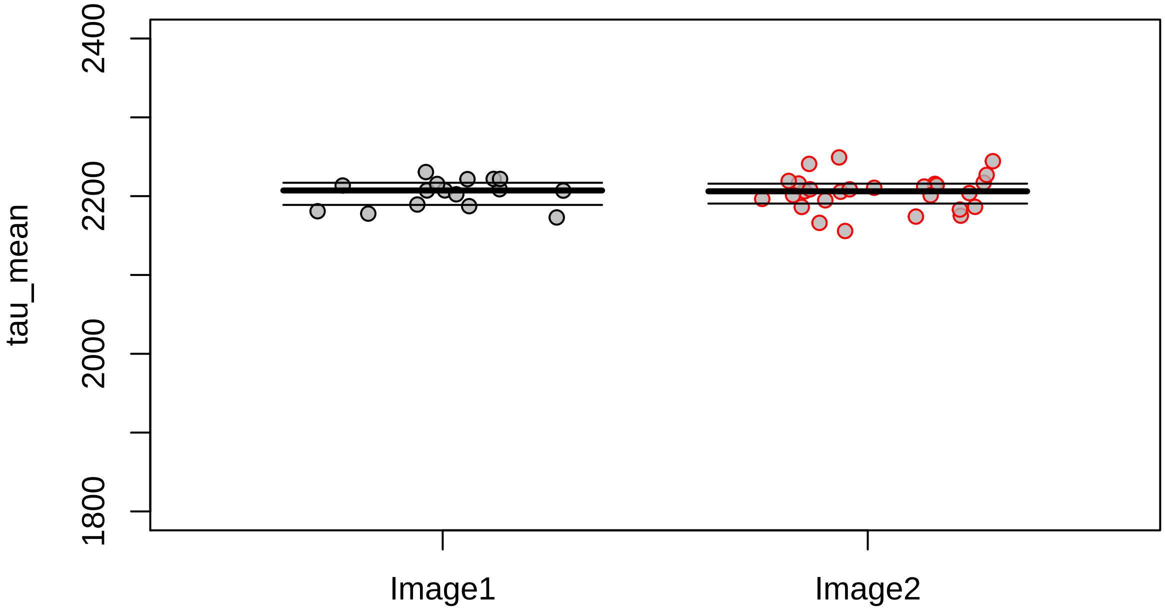
We can repeat the same procedure for a second FLIM image

```
tauFilePath2 <-  paste(path, "PAS406_002_t1.asc", sep="")
photFilePath2 <-  paste(path, "PAS406_002_photons.asc", sep="")

data2 <- getData(pathTau = tauFilePath2, pathPhoton = photFilePath2, label = "Image2")
res2 <- dbscan(data2[,c("row","col")], eps = 1, minPts = 3)
data2$bacteria <- res2$cluster

Im2_sumup <- data2 %>% filter(bacteria != 0) %>%
  group_by(bacteria) %>%
  dplyr::summarise( nPixels =n(),
             tau_mean = mean(tau1d, na.rm = T),
             tau_sd = sd(tau1d, na.rm = T),
             tau_med = median(tau1d, na.rm = T),
             label = unique(ind)) %>%
  filter(nPixels<80)

#Im2_sumup
```

Let’s bind table 1 and table 2 to check for example visually if we obtained the same results and lifetime values in these two images.

```
finalTable <- bind_rows( Im1_sumup, Im2_sumup)
utilitR::stripPlot(data = finalTable, tau_mean ~ label, addBoxplot = T, ylim=c(1800, 2400))
```

```
## Loading required package: beeswarm
```



In this plot, each point corresponds to the average lifetime of a bacteria measured either in image 1 or in image 2. Horizontal lines correspond to the median value of cells (bold line) and the interquartile range (IQR). The lifetimes obtained for the two images look very similar. Indeed, a null hypothesis statistical test fails to demonstrate any difference of the bacteria cell average lifetime in these two images - confirming that reproducible results can be easily obtained in between independant experiements realized at different dates of experiments.

```
summary(glm(tau_mean ~ label, data = finalTable, family = "gaussian"))
```

```
##
## Call:
## glm(formula = tau_mean ~ label, family = "gaussian", data = finalTable)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -48.64  -15.76    3.09   11.47   44.72
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2204.1062     5.2099 423.059   <2e-16 ***
## labelImage2    0.3196     6.5748   0.049    0.961
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 434.2927)
##
##     Null deviance: 17807  on 42  degrees of freedom
## Residual deviance: 17806  on 41  degrees of freedom
## AIC: 387.15
##
## Number of Fisher Scoring iterations: 2
```

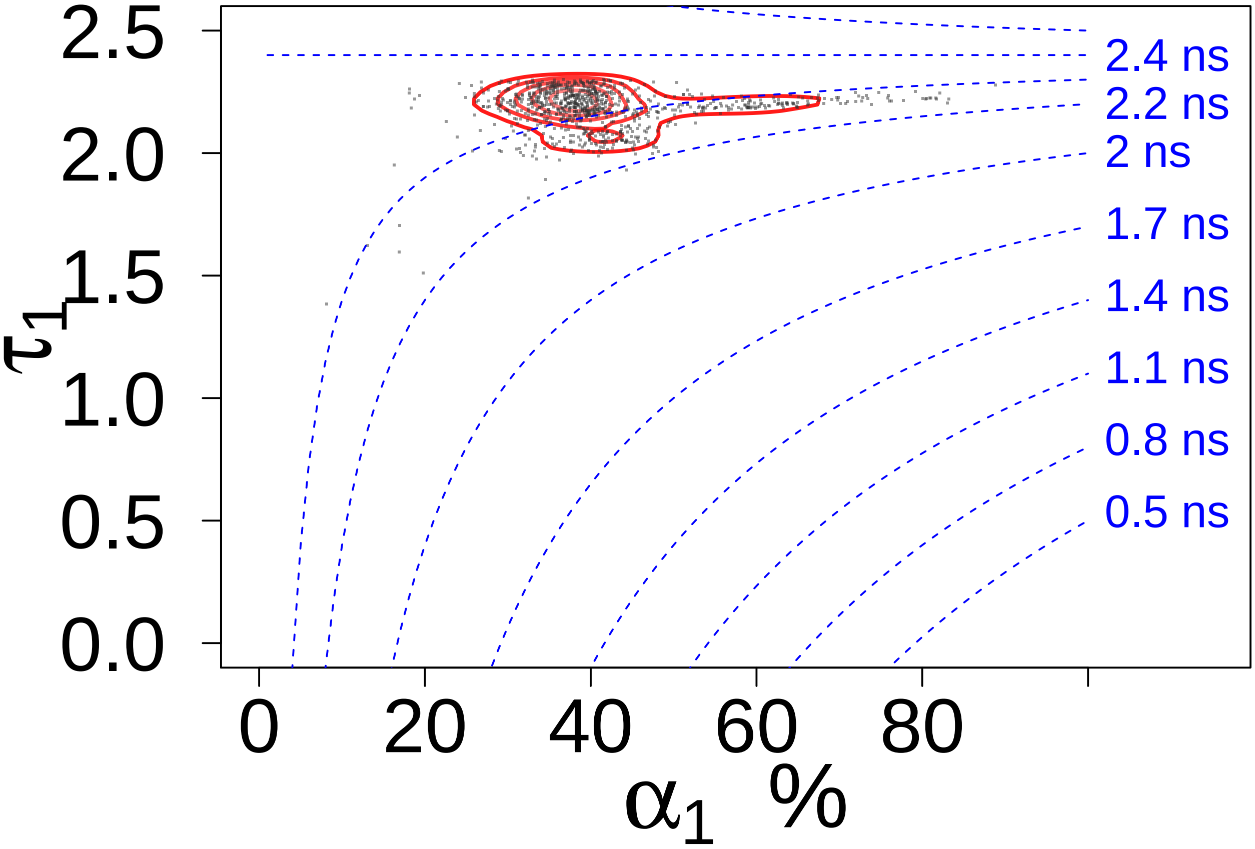
Advanced data exploration for double-explonential fit

First upload the data corresponding to double-explonential fit and write them locally.

```
download.file(url = paste(path, "407_20170302_002_t1.asc", sep=""),
              destfile = file.path(getwd(),"407_20170302_002_t1.asc"))
download.file(url = paste(path, "407_20170302_002_a1%5B%25%5D.asc", sep=""),
              destfile = file.path(getwd(),"407_20170302_002_a1[%].asc"))
download.file(url = paste(path, "407_20170302_002_a2%5B%25%5D.asc", sep=""),
              destfile = file.path(getwd(),"407_20170302_002_a2[%].asc"))
```

Then to plot the FLIM diagram, two lines are necessary. First load and read the data, then render the plot.

```
pathlowFret <- file.path(getwd(),  "407_20170302_002")
qsD <- getDataD(pathEGFP = pathlowFret, label = "PAS407" , ref = 2300)#load
plotDiagram(data = qsD, silence =TRUE)
```



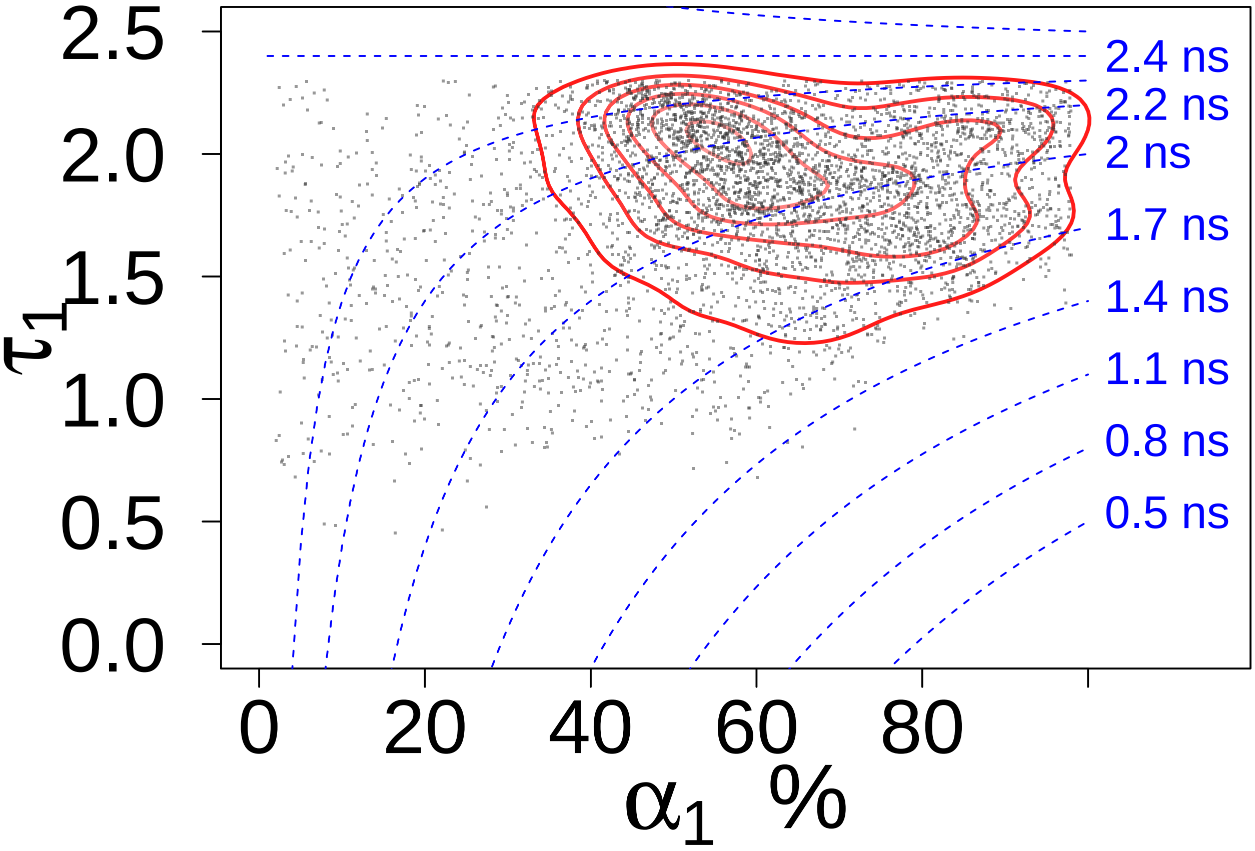
This plot corresponds to a situation for which the number of donors is much higher than the number of acceptors. The fraction of transfering species is low (about 25-30 %). The lifetime values distribution is narrow with τ_1 values centerer at about 2.3 ns.

If we now look at data corresponding to a strain with the opposite labelling strategy (the donor is now on the lowest expressed enzyme), the diagram plot looks very different.

```
download.file(url = paste(path, "476_20170428_001_t1.asc", sep=""),
              destfile = file.path(getwd(),"476_20170428_001_t1.asc"))
download.file(url = paste(path, "476_20170428_001_a1%5B%25%5D.asc", sep=""),
              destfile = file.path(getwd(),"476_20170428_001_a1[%].asc"))
download.file(url = paste(path, "476_20170428_001_a2%5B%25%5D.asc", sep=""),
              destfile = file.path(getwd(),"476_20170428_001_a2[%].asc"))
```

```
download.file(url = paste(path, "476_20170428_002_t1.asc", sep=""),
              destfile = file.path(getwd(),"476_20170428_002_t1.asc"))
download.file(url = paste(path, "476_20170428_002_a1%5B%25%5D.asc", sep=""),
              destfile = file.path(getwd(),"476_20170428_002_a1[%].asc"))
download.file(url = paste(path, "476_20170428_002_a2%5B%25%5D.asc", sep=""),
              destfile = file.path(getwd(),"476_20170428_002_a2[%].asc"))
```

```
pathhighFret1 <- paste(getwd(),  "476_20170428_001", sep="/")
pathhighFret2 <- paste(getwd(),  "476_20170428_002", sep="/")
qsF1 <- getDataF(pathFRET =pathhighFret1, label = "PAS476" )#load
qsF2 <- getDataF(pathFRET =pathhighFret2, label = "PAS476" )#load
qsF <- bind_rows(qsF1, qsF2)
plotDiagram(data = qsF, silence =TRUE,nl = 8)
```



Here, α_1 values are much higher than in the previous plot, ranging from 40% to nearly 100%. The large distribution of lifetimes shows the coexistence of complexes with different numbers of acceptors and different FRET efficiencies.

The graphical analysis of the same complexe but with two different labelling order demonstrates how the FLIM diagram can find very interesting visualization properties regarding complexes with unbalanced stoichiometry - a situation usually challenging to interpretate in FLIM-FRET data.