

# Umbrella Tutorial 2D

*Bart K. M. Jacobs*

*7 april 2017*

---

This tutorial explains how to run Umbrella when multi-channel data are available in an RData file in the correct format.

To create an RData file from .csv files, please consult [Tutorial\\_DataCreation](#).

To analyse single-channel data, please consult [Tutorial\\_Umbrella\\_2D](#).

## Introduction to R

First, load the RData file `Tutorial_2D.RData`. You can do this through File -> Load Workspace in Base R, or File -> Open File... in RStudio. Alternatively, execute the following command after changing the path accordingly

```
load("C:\\temp\\Tutorial_2D.RData")
```

With the `ls()` command, you can see all objects currently loaded.

In RStudio, such list can also be consulted in the Global Environment, in the upper right panel.

```
ls()
```

```
## [1] "PlotUmbrella2d"    "Tutorial.Plate.2D" "Umbrella2d"
```

This file contains three objects:

- “Tutorial.Plate.2D”, a simulated example dataset
- “Umbrella2d”, the Umbrella function for multichannel data
- “PlotUmbrella2d”, a routine to create graphics

You can check the structure of an R object with `str()` and `attributes()`

```
str(Tutorial.Plate.2D)
```

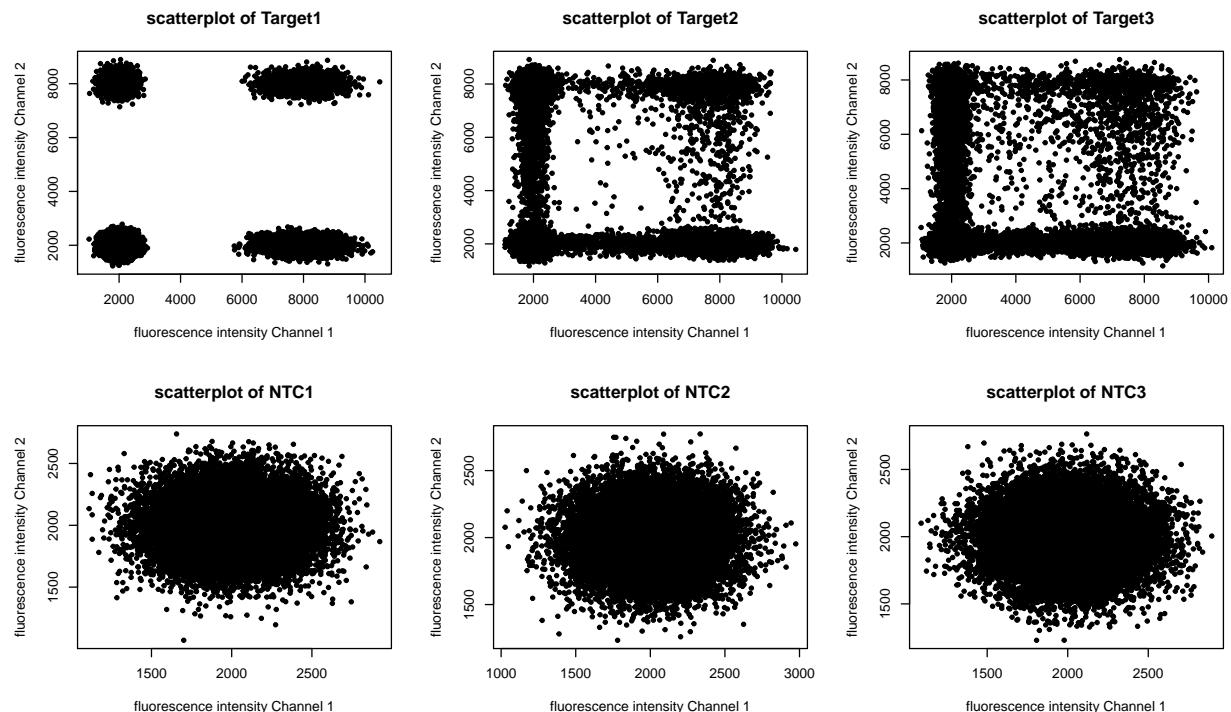
```
## List of 6
## $ Target1:'data.frame': 20000 obs. of  2 variables:
##   ..$ Ch1.Amplitude: num [1:20000] 2086 1821 1744 2198 8712 ...
##   ..$ Ch2.Amplitude: num [1:20000] 2000 8079 8130 2191 2011 ...
## $ Target2:'data.frame': 20000 obs. of  2 variables:
##   ..$ Ch1.Amplitude: num [1:20000] 1827 8848 1710 2018 1530 ...
##   ..$ Ch2.Amplitude: num [1:20000] 7012 2249 1594 8209 8029 ...
## $ Target3:'data.frame': 20000 obs. of  2 variables:
##   ..$ Ch1.Amplitude: num [1:20000] 2139 1720 2018 2376 7485 ...
##   ..$ Ch2.Amplitude: num [1:20000] 1666 1897 7974 2106 6326 ...
## $ NTC1   :'data.frame': 20000 obs. of  2 variables:
##   ..$ Ch1.Amplitude: num [1:20000] 1919 1843 2268 2334 2020 ...
##   ..$ Ch2.Amplitude: num [1:20000] 2119 1757 2315 1829 2280 ...
## $ NTC2   :'data.frame': 20000 obs. of  2 variables:
##   ..$ Ch1.Amplitude: num [1:20000] 2159 2012 2182 1485 2086 ...
##   ..$ Ch2.Amplitude: num [1:20000] 2105 1694 1759 1889 2155 ...
## $ NTC3   :'data.frame': 20000 obs. of  2 variables:
```

```
## ...$ Ch1.Amplitude: num [1:20000] 1725 1989 1763 2270 2042 ...
## ...$ Ch2.Amplitude: num [1:20000] 1904 1915 1841 1809 1659 ...
```

Input data for Umbrella needs to be structured as a list with a single, named entry for each partition set.

Let's make scatterplots of the data:

```
par(mfrow=c(2,3))
for(i in 1:6){
  plot(Tutorial.Plate.2D[[i]], pch=20, xlab="fluorescence intensity Channel 1",
       ylab="fluorescence intensity Channel 2", main=paste("scatterplot of", names(Tutorial.Plate.2D)[i]))
}
```



```
par(mfrow=c(1,1))
```

We see that the clusters are nicely separated in Target 1, while Target2 and Target3 show increasing rain.

## Running Umbrella

In general, you want to open the `Umbrella_2D.R` file and run all the code in it. Note that this file will have a version number which can change as the software gets updated: e.g. `Umbrella_2d_V0.R`

For this tutorial, the functions have already been included in the Rdata file.

Umbrella requires 5 other packages to be loaded, these are: MASS, mgcv, robust, modeest, OrdMonReg

Please intall them before running the procedure. This can be done through Packages -> Install Package(s)... in Base R or with Tools -> Install Packages... in RStudio. In most installations, running the following code will work as well:

```
install.packages(c("MASS", "mgcv", "robust", "modeest", "OrdMonReg"))
```

Run Umbrella on the tutorial data set with the following code:

```

ResTut <- Umbrella2d(Tutorial.Plate.2D, NTC=c("NTC1", "NTC2", "NTC3"))

##      Warning: not checking for rotation can lead to wrong results if rotation is present.
## Preprocessing complete
## NTC identification complete
## Estimation Target1 completed in 51.74517 secs .
##
## Well Target1 :
##
##   Channel 1:
## 20000 partitions
## Robust estimator:
##   6023 positive, p = 0.3012 ( CI: [ 0.312 , 0.2903 ] )
##   421.6 copies/mul ( CI: [ 400.5 , 442.7 ] )
## Threshold estimator:
##   6021 positive, p = 0.301 ( CI: [ 0.301 , 0.301 ] )
##   421.4 copies/mul ( CI: [ 410.7 , 432.1 ] )
##
##   Channel 2:
## 20000 partitions
## Robust estimator:
##   5940 positive, p = 0.297 ( CI: [ 0.3005 , 0.2935 ] )
##   414.6 copies/mul ( CI: [ 402.5 , 426.6 ] )
## Threshold estimator:
##   5969 positive, p = 0.2984 ( CI: [ 0.2984 , 0.2984 ] )
##   417 copies/mul ( CI: [ 406.4 , 427.7 ] )
##
## CI's for concentration include Poisson variability.
##
## -----
## Estimation Target2 completed in 48.77388 secs .
##
## Well Target2 :
##
##   Channel 1:
## 20000 partitions
## Robust estimator:
##   6168 positive, p = 0.3084 ( CI: [ 0.3196 , 0.2973 ] )
##   433.9 copies/mul ( CI: [ 412 , 455.7 ] )
## Threshold estimator:
##   5897 positive, p = 0.2948 ( CI: [ 0.2936 , 0.2962 ] )
##   411 copies/mul ( CI: [ 400.2 , 421.8 ] )
##
##   Channel 2:
## 20000 partitions
## Robust estimator:
##   5943 positive, p = 0.2972 ( CI: [ 0.3083 , 0.2861 ] )
##   414.9 copies/mul ( CI: [ 393.5 , 436.3 ] )
## Threshold estimator:
##   5907 positive, p = 0.2954 ( CI: [ 0.2948 , 0.296 ] )
##   411.8 copies/mul ( CI: [ 401.2 , 422.5 ] )
##
## CI's for concentration include Poisson variability.
##

```

```

## -----
## Estimation Target3 completed in 54.56846 secs .
##
## Well Target3 :
##
##   Channel 1:
## 20000 partitions
## Robust estimator:
##   5981 positive, p = 0.2991 ( CI: [ 0.3114 , 0.2868 ] )
##   418 copies/mul ( CI: [ 394.8 , 441.3 ] )
## Threshold estimator:
##   5715 positive, p = 0.2858 ( CI: [ 0.2834 , 0.288 ] )
##   395.9 copies/mul ( CI: [ 384.7 , 407.1 ] )
##
##   Channel 2:
## 20000 partitions
## Robust estimator:
##   5686 positive, p = 0.2843 ( CI: [ 0.2962 , 0.2724 ] )
##   393.5 copies/mul ( CI: [ 371.4 , 415.6 ] )
## Threshold estimator:
##   5730 positive, p = 0.2865 ( CI: [ 0.2856 , 0.2872 ] )
##   397.1 copies/mul ( CI: [ 386.7 , 407.6 ] )
##
## CI's for concentration include Poisson variability.
##
## -----
## Estimation NTC1 completed in 37.79578 secs .
##
## Well NTC1 :
##
##   Channel 1:
## 20000 partitions
## Robust estimator:
##   0 positive, p = 0 ( CI: [ 0 , 0 ] )
##   0 copies/mul ( CI: [ 0 , 0 ] )
## Threshold estimator:
##   0 positive, p = 0 ( CI: [ 0 , 0 ] )
##   0 copies/mul ( CI: [ 0 , 0 ] )
##
##   Channel 2:
## 20000 partitions
## Robust estimator:
##   77 positive, p = 0.0038 ( CI: [ 0.0127 , 0 ] )
##   4.5 copies/mul ( CI: [ 0 , 15.1 ] )
## Threshold estimator:
##   0 positive, p = 0 ( CI: [ 0 , 0 ] )
##   0 copies/mul ( CI: [ 0 , 0 ] )
##
## CI's for concentration include Poisson variability.
##
## -----
## Estimation NTC2 completed in 28.11881 secs .
##
## Well NTC2 :

```

```

## 
##   Channel 1:
## 20000 partitions
## Robust estimator:
##    79 positive, p = 0.0039 ( CI: [ 0.0121 , 0 ] )
##    4.6 copies/mul ( CI: [ 0 , 14.4 ] )
## Threshold estimator:
##    0 positive, p = 0 ( CI: [ 0 , 0 ] )
##    0 copies/mul ( CI: [ 0 , 0 ] )
##
##   Channel 2:
## 20000 partitions
## Robust estimator:
##    4 positive, p = 2e-04 ( CI: [ 0.0058 , 0 ] )
##    0.3 copies/mul ( CI: [ 0 , 6.9 ] )
## Threshold estimator:
##    0 positive, p = 0 ( CI: [ 0 , 0 ] )
##    0 copies/mul ( CI: [ 0 , 0 ] )
##
## CI's for concentration include Poisson variability.
##
## -----
## Estimation NTC3 completed in 45.07551 secs .
##
## Well NTC3 :
##
##   Channel 1:
## 20000 partitions
## Robust estimator:
##    0 positive, p = 0 ( CI: [ 0.0023 , 0 ] )
##    0 copies/mul ( CI: [ 0 , 3.7 ] )
## Threshold estimator:
##    0 positive, p = 0 ( CI: [ 0 , 0 ] )
##    0 copies/mul ( CI: [ 0 , 0 ] )
##
##   Channel 2:
## 20000 partitions
## Robust estimator:
##    0 positive, p = 0 ( CI: [ 0 , 0 ] )
##    0 copies/mul ( CI: [ 0 , 0 ] )
## Threshold estimator:
##    0 positive, p = 0 ( CI: [ 0 , 0 ] )
##    0 copies/mul ( CI: [ 0 , 0 ] )
##
## CI's for concentration include Poisson variability.
##
## -----
## Estimation complete

```

Only the first argument is needed. This should be a list of partition sets with each having two channel fluorescence intensities.

Currently, there are four optional arguments:

- vol= for the volume in nanoliter, default value 0.85 for Bio-Rad's QX100.

- NTC= for a vector with names of the NTCs.
- Plate= find cluster center using information from all partitions sets or not. If true, all plate information may be used to calculate cluster centres (default). If false, only cluster information (if given) and individual partition sets are used.
- Rotate= how to correct for rotation/tilt of the clusters. If false, no data rotation is performed. (default). If true, data gets rotated based on a default, partition set-based algorithm. If “Plate”, data gets rotated based on data from complete plate. If a name of one or multiple partition sets is given, all partition sets are rotated based on data from this (these) reference(s).

In this example, we keep the default value for the volume, plate and rotate, while supplying three NTCs with names NTC1, NTC2, NTC3.

Note: Umbrella can run without NTCs, but underlying distributional assumptions are likely violated in this case! Please do provide NTCs.

Extensive output is stored in the “ResTut” object. The name of the result object can be freely chosen.

While the procedure is running, some output is already shown in the R console. This includes:

- During the first run: messages about all packages that are loaded.
- The steps when they are concluded.
- Per partition set:
  - The time to complete the estimation step.
  - The number of partitions (for each channel).
- The number of positives and the concentration for both the overall Umbrella estimator and a probabilistic threshold estimator (for each channel).
- A confidence interval for this number and the concentration (for each channel).

A new object, in this case named “ResTut” is created.

The structure of this object can be summarized with `str()`. The object is a list with a separate entry for each partition set. The structure is the same over all partition sets. In what follows, we’ll look at the details for Target3.

```
str(ResTut$Target3,max.level=1)
```

```
## List of 13
## $ conc    :List of 2
## $ thresh   :List of 2
## $ fits     :List of 4
## $ data     :'data.frame':   20000 obs. of  3 variables:
## $ droppi   : num [1:20000, 1:2] 9.94e-01 1.00 1.00 9.40e-01 2.86e-12 ...
## ..- attr(*, "dimnames")=List of 2
## $ dropci   :List of 2
## $ reppi    :List of 2
## $ densfits:List of 2
## $ ntcfits  :List of 2
## $ pifits   :List of 2
## $ cluspi   : num [1:20000, 1:4] 9.94e-01 1.00 1.29e-12 9.40e-01 1.22e-23 ...
## $ name     : chr "Target3"
## $ rawdata  :'data.frame':   20000 obs. of  2 variables:
```

For each partition set, a list of 11 pieces of information is created. In what follows, we’ll go through all of them, although most likely you will only want to use the first few.

1. \$ conc

```
ResTut$Target3$conc$x
```

```
##      prob_robust conc_robust prob_thres conc_thres
## est    0.299057375 418.034404 0.285750000 395.908515
## sd     0.005431597  9.116458 0.001145644  1.887037
## CI LB 0.311350011 394.815732 0.283450000 384.746286
## CI UB 0.286764740 441.253075 0.287950000 407.070744
```

```
ResTut$Target3$conc$y
```

```
##      prob_robust conc_robust prob_thres conc_thres
## est    0.28431070 393.540176 0.286500000 397.1445201
## sd     0.00525821  8.643596 0.000425245  0.7011749
## CI LB 0.29621093 371.443173 0.285600000 386.6914824
## CI UB 0.27241047 415.637179 0.287250000 407.5975578
```

For each of the channels separately: These are the estimates of (i) the probability to be positive and (ii) the concentration of the sample (in copies/microliter) for both Umbrella estimators (overall and probability threshold). (first line) Additionally, their standard deviation (second line) and confidence intervals are added (third and fourth line).

Note: proper standard deviations and confidence intervals can only be calculated when multiple NTCs are provided

2. \$ thres

```
ResTut$Target3$tresh$x
```

```
##      prob CI low p CI high p count CI low n CI high n
## pos   0.28575 0.28345  0.28795 5715      5669      5759
## neg   0.70595 0.70415  0.70755 14119     14083     14151
## rain  0.00830 0.00475  0.01215 166       95       243
```

```
ResTut$Target3$tresh$y
```

```
##      prob CI low p CI high p count CI low n CI high n
## pos   0.28650 0.28560  0.28725 5730      5712      5745
## neg   0.70575 0.70345  0.70810 14115     14069     14162
## rain  0.00775 0.00465  0.01095 155       93       219
```

For each of the channels separately: These are the proportion (probability) of partitions that are positive (pos), negative (neg) or rain (rain) with a confidence interval on each, as well as the associated absolute numbers.

Proportions are calculated as absolute numbers divided by the total number of partitions.

3. \$ fits

```
ResTut$Target3$fits
```

```
## $mu
##      [,1]      [,2]
## xMode00 2001.752 2009.587
## xMode00 2001.752 2009.587
## xMode00 2001.752 2009.587
##
## $sig
##      [,1]      [,2]
## 248.1197 200.9678
## 250.3868 199.6059
## 246.7626 199.5138
```

```

##  

## $pi  

##      [,1]      [,2]  

## 0.6952996 0.7216142  

## 0.7061347 0.7138761  

## 0.7013936 0.7115776  

##  

## $sigdif  

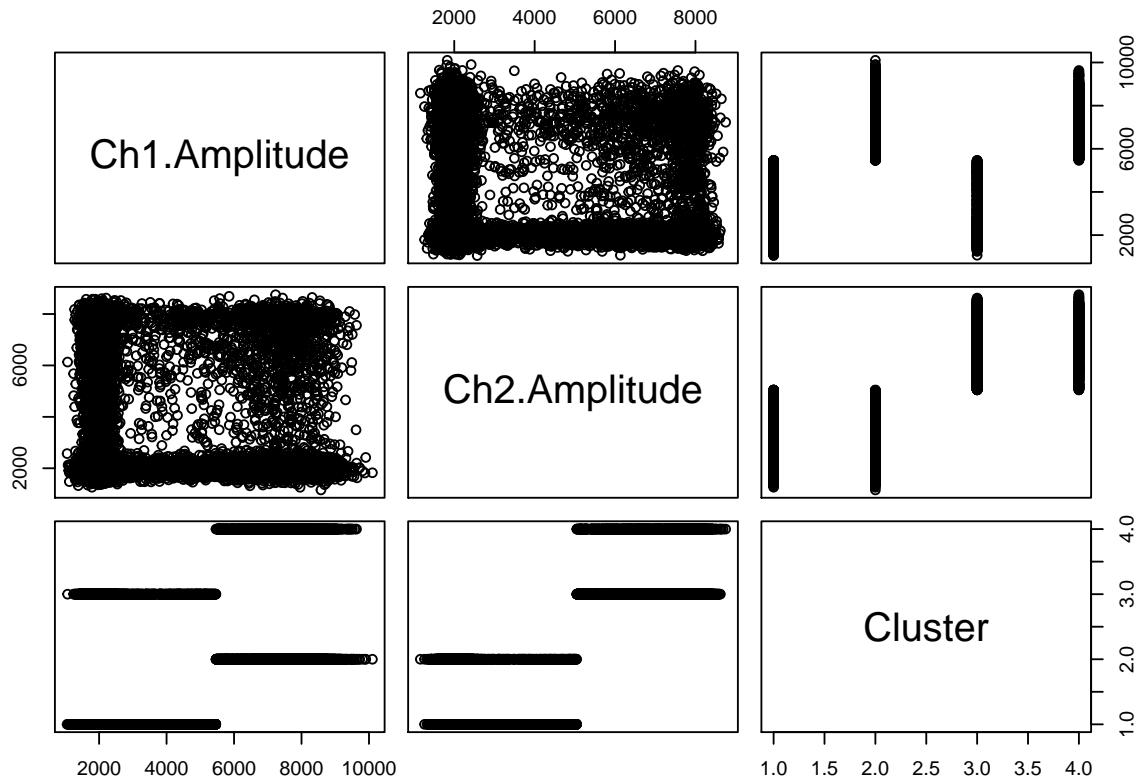
## [1] 1.000000 1.044875

```

Estimates used for the mode (mu), robust deviation (sig) and the probability that a random target partition is negative (pi) for both channels. These estimates are recalculated for each combination with a different NTC. I.e.: the first number refers to the estimate calculated and used in the joint model with Target3 and NTC1, the second number is the estimate from the model with Target3 and NTC2, and so on. Additionally, the ratio of the variation in the single positive clusters compared to the double negative cluster.

4. \$ data

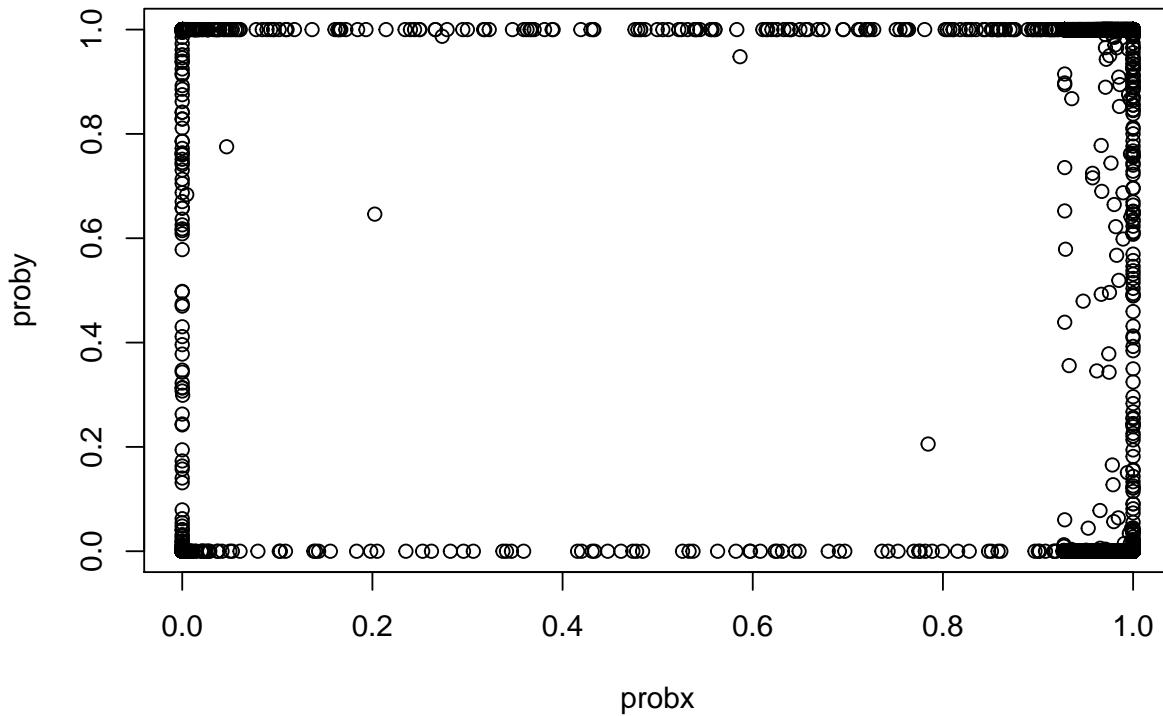
```
plot(ResTut$Target3$data)
```



The original data with a clustering (preprocessed clustering, not final!) added.

5. \$ droppi

```
plot(ResTut$Target3$droppi)
```



For each of the channels separately: The estimates for the individual partition probabilities

6. \$ dropci

```
str(ResTut$Target3$dropci)
```

```
## List of 2
## $ x: num [1:20000, 1:2] 0.89 1 0.956 0.638 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   ... $ : NULL
##   ... $ : chr [1:2] "probciLx" "probciUx"
## $ y: num [1:20000, 1:2] 1 0.955 0 0.955 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   ... $ : NULL
##   ... $ : chr [1:2] "probciLy" "probciUy"
```

For each of the channels separately: Confidence intervals for the individual droplet probabilities

7. \$ reppi

```
str(ResTut$Target3$reppi)
```

```
## List of 2
## $ x: num [1:20000, 1:3] 9.71e-01 1.00 1.02 9.54e-01 1.64e-12 ...
## $ y: num [1:20000, 1:3] 1.00 9.89e-01 1.09e-12 1.02 3.85e-12 ...
```

For each of the channels separately: The original partition probabilities per NTC before antitonic regression and averaging over NTCs. Mainly used for quality control (QC).

#### 8. \$ densfits

```
str(ResTut$Target3$densfits)
```

```
## List of 2
## $ x:List of 3
## ...$ : Named num [1:20000] 9.73e-04 6.20e-04 1.10e-03 3.81e-04 9.43e-05 ...
## ... - attr(*, "names")= chr [1:20000] "7091" "1309" "2154" "9389" ...
## ...$ : Named num [1:20000] 9.74e-04 6.13e-04 1.10e-03 3.81e-04 9.42e-05 ...
## ... - attr(*, "names")= chr [1:20000] "7091" "1309" "2154" "9389" ...
## ...$ : Named num [1:20000] 9.77e-04 6.14e-04 1.10e-03 3.85e-04 9.45e-05 ...
## ... - attr(*, "names")= chr [1:20000] "7091" "1309" "2154" "9389" ...
## $ y:List of 3
## ...$ : Named num [1:20000] 3.57e-04 1.22e-03 1.46e-04 1.26e-03 4.16e-05 ...
## ... - attr(*, "names")= chr [1:20000] "467" "2949" "13652" "7068" ...
## ...$ : Named num [1:20000] 3.51e-04 1.21e-03 1.44e-04 1.26e-03 4.15e-05 ...
## ... - attr(*, "names")= chr [1:20000] "467" "2949" "13652" "7068" ...
## ...$ : Named num [1:20000] 3.52e-04 1.21e-03 1.44e-04 1.26e-03 4.15e-05 ...
## ... - attr(*, "names")= chr [1:20000] "467" "2949" "13652" "7068" ...
```

For each of the channels separately: The fitted density of the target when jointly modelled with each NTC.

#### 9. \$ ntcfits

```
str(ResTut$Target3$ntcfits)
```

```
## List of 2
## $ x:List of 3
## ...$ : Named num [1:20000] 9.44e-04 5.92e-04 1.12e-03 3.63e-04 1.54e-16 ...
## ... - attr(*, "names")= chr [1:20000] "7091" "1309" "2154" "9389" ...
## ...$ : Named num [1:20000] 9.73e-04 5.97e-04 1.12e-03 3.79e-04 1.57e-16 ...
## ... - attr(*, "names")= chr [1:20000] "7091" "1309" "2154" "9389" ...
## ...$ : Named num [1:20000] 9.88e-04 5.85e-04 1.12e-03 3.35e-04 1.56e-16 ...
## ... - attr(*, "names")= chr [1:20000] "7091" "1309" "2154" "9389" ...
## $ y:List of 3
## ...$ : Named num [1:20000] 3.49e-04 1.21e-03 1.60e-16 1.29e-03 1.60e-16 ...
## ... - attr(*, "names")= chr [1:20000] "467" "2949" "13652" "7068" ...
## ...$ : Named num [1:20000] 3.33e-04 1.22e-03 1.59e-16 1.27e-03 1.59e-16 ...
## ... - attr(*, "names")= chr [1:20000] "467" "2949" "13652" "7068" ...
## ...$ : Named num [1:20000] 3.32e-04 1.22e-03 1.58e-16 1.26e-03 1.58e-16 ...
## ... - attr(*, "names")= chr [1:20000] "467" "2949" "13652" "7068" ...
```

For each of the channels separately: The fitted density of each NTC when jointly modelled with the target.

#### 10. \$ pifits

```
str(ResTut$Target3$pifits)
```

```
## List of 2
## $ x:List of 3
## ...$ : num [1:51(1d)] 0.823 0.798 0.776 0.755 0.736 ...
## ... - attr(*, "dimnames")=List of 1
## ... ...$ : chr [1:51] "52" "53" "54" "55" ...
## ...$ : num [1:51(1d)] 0.716 0.714 0.713 0.711 0.71 ...
## ... - attr(*, "dimnames")=List of 1
## ... ...$ : chr [1:51] "52" "53" "54" "55" ...
```

```

##   ..$ : num [1:51(1d)] 0.701 0.706 0.709 0.709 0.707 ...
##   ... -- attr(*, "dimnames")=List of 1
##   ... .$. : chr [1:51] "52" "53" "54" "55" ...
## $ y:List of 3
##   ..$ : num [1:51(1d)] 0.73 0.767 0.799 0.824 0.841 ...
##   ... -- attr(*, "dimnames")=List of 1
##   ... .$. : chr [1:51] "52" "53" "54" "55" ...
##   ..$ : num [1:51(1d)] 0.631 0.653 0.674 0.693 0.711 ...
##   ... -- attr(*, "dimnames")=List of 1
##   ... .$. : chr [1:51] "52" "53" "54" "55" ...
##   ..$ : num [1:51(1d)] 0.496 0.526 0.557 0.589 0.619 ...
##   ... -- attr(*, "dimnames")=List of 1
##   ... .$. : chr [1:51] "52" "53" "54" "55" ...

```

For each of the channels separately: Ratio between target and NTC around the mode. Can be used for QC.

11. \$ cluspi

```
str(ResTut$Target3$cluspi)
```

```
## num [1:20000, 1:4] 9.94e-01 1.00 1.29e-12 9.40e-01 1.22e-23 ...
```

The estimates for the individual partition probabilities per cluster rather than per channel: i.e.: the probability to be double negative, single negative on channel 1, single negative on channel 2, double positive.

12. \$ name

Name of the partition set, mainly used for internal automatic calling and matching and when displaying the results.

## Summarizing Output

### Automatic Figures

We created a file to provide some useful figures automatically. To use this, you generally want to open **Graphics\_Umbrella\_2d.R** and execute all the code. For this tutorial, we included the function in the already loaded RData file.

Change the path below to the folder where you want figures to appear and then run the code.

```
setwd("C:/temp")
```

Run the code below to generate and save some graphics directly onto your hard disk.

```
PlotUmbrella2d(ResTut, "Tutorial2D")
```

```
## pdf
## 2
```

The command has three arguments of which two are required.

- The name of an object created by the Umbrella2d procedure (here: ResTut)
- A name for the files that can be freely chosen (here: Tutorial2D)
- Optionally, the resolution can be chosen with the **res=** command (default 120 dpi)

For each partition set, two summary plots will be created. The first one consists of two figures similar to the middle panel of Figure 2 of our manuscript, one for each channel (univariate summary). The second shows the scatterplot of the data three times, once as the original data, and twice after rotation and coloured with

probabilities per channel. Important numeric output is added to this plot as well. Finally, a single QC figure is added (all partition sets together).

As an example, for Target3, Figure 1 shows the univariate output, while Figure 2 shows the multivariate graphs.

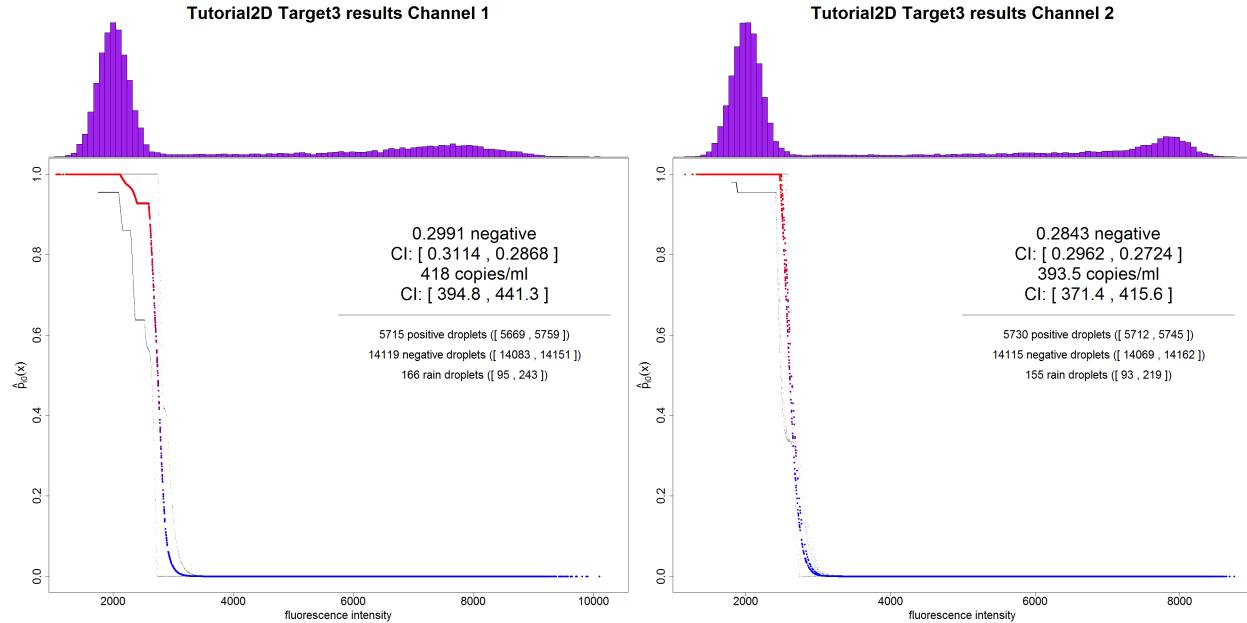


Figure 1: Univariate output of PlotUmbrella2d

Using the R object (ResTut here), it is possible to create custom output. We give a few handy examples similar to figures in the manuscript.

### Table with concentration estimates

Creating a table with concentration estimates for the complete plate / experiment can be done with the following code:

```
conrobx <- t(sapply(ResTut,function(PS){PS$conc$x[c(1,3,4),2]}))
conrobx
```

	est	CI LB	CI UB
## Target1	421.578842	400.4571	442.700567
## Target2	433.856641	412.0000	455.713316
## Target3	418.034404	394.8157	441.253075
## NTC1	0.000000	0.0000	0.000000
## NTC2	4.633868	0.0000	14.377544
## NTC3	0.000000	0.0000	3.720962

The table above displays the results of the main Umbrella estimator for the first channel.

```
conptrx <- t(sapply(ResTut,function(PS){PS$conc$x[c(1,3,4),4]}))
conptrx
```

	est	CI LB	CI UB
## Target1	421.3836	410.6830	432.0843
## Target2	410.9938	400.1665	421.8211

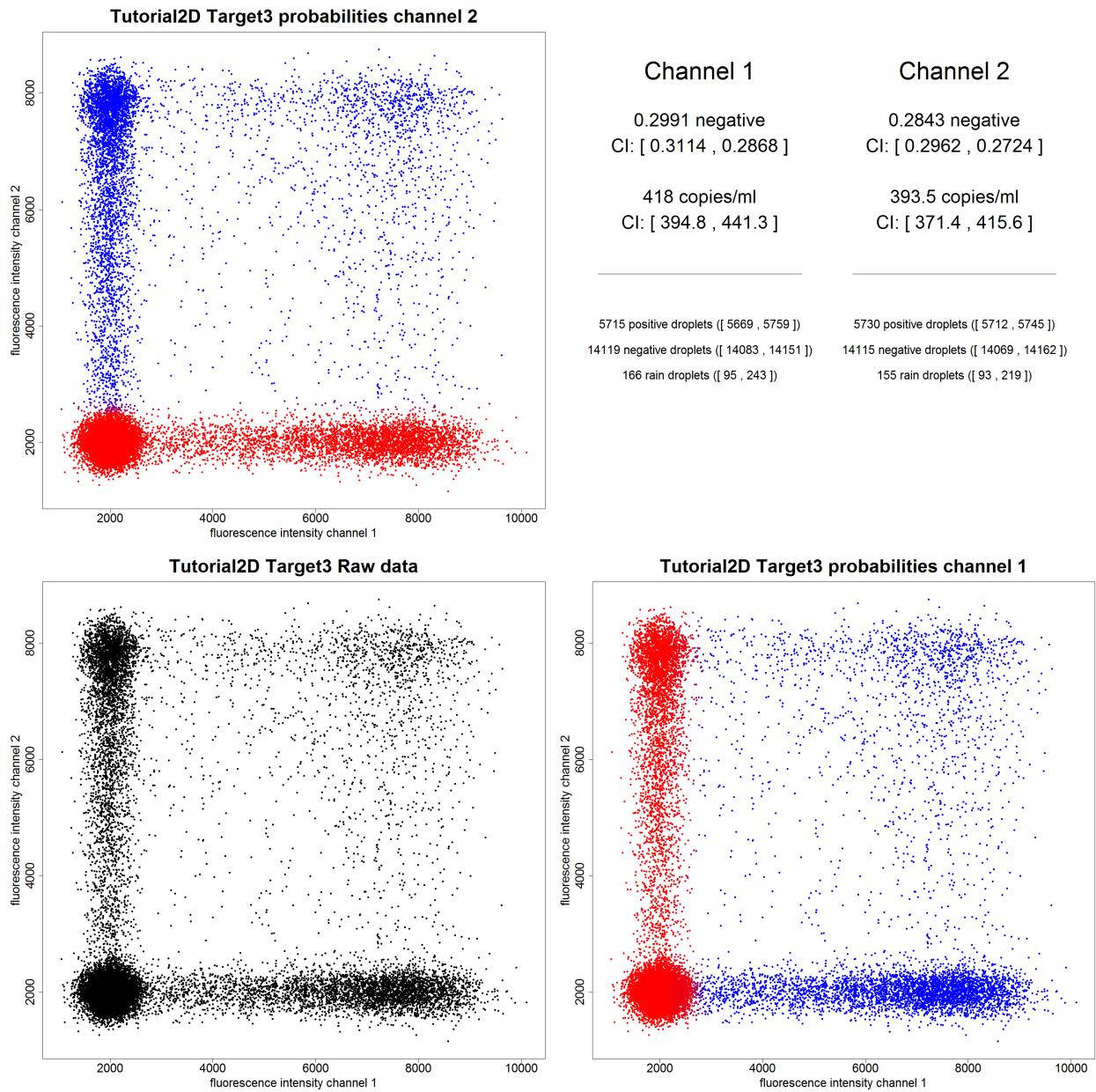


Figure 2: Multivariate output of PlotUmbrella2d

```

## Target3 395.9085 384.7463 407.0707
## NTC1      0.0000  0.0000  0.0000
## NTC2      0.0000  0.0000  0.0000
## NTC3      0.0000  0.0000  0.0000

```

This table shows the results of the probability threshold estimator for the first channel.

Results for the second channel can be obtained by replacing x by y or joining both.

We can now save these together in a handy comma separated value file, which can be opened by most data processing programs, but also in Excel, Notepad, etc.

```

concdat <- rbind(t(sapply(ResTut,function(PS){PS$conc$x[c(1,3,4),c(2,4)]})),
                  t(sapply(ResTut,function(PS){PS$conc$y[c(1,3,4),c(2,4)]})))
colnames(concdat) <- c("est_main","CI_LB_main","CI_UB_main","est_thres","CI_LB_thres","CI_UB_thres")
write.csv(data.frame(PartSet=rownames(concdat),Channel=rep(c("Ch1","Ch2"),each=6),concdat),
          file="Tutorial_Concentrations.csv",row.names=F)

```

## Plotting the concentrations

This creates output similar to Figure 3 in the manuscript, now with two channels plotted next to each other rather than two different methods.

```

conroby <- t(sapply(ResTut,function(PS){PS$conc$y[c(1,3,4),2]}))
ymaxs <- max(conrobx[conrobx<Inf],conroby[conroby<Inf])
npset <- nrow(conrobx)
par(mar=c(1,4,1,1))
plot(NULL,ylim=c(0,ymaxs),xlim=c(0,npset+1),
      main="",xlab="",xaxt="n",
      ylab=expression(paste("concentration (cp/",mu,"l)",sep="")))
for(i in 1:npset){
  lines(c(i-0.1,i-0.1),c(conrobx[i,2],conrobx[i,3]))
  lines(c(i+0.1,i+0.1),c(conroby[i,2],conroby[i,3]))
  points(i-0.1,conrobx[i,1],col="blue",pch=16)
  points(i+0.1,conroby[i,1],col="green",pch=16)
}

```

