Package: popPCR (via r-universe)

October 5, 2026

Title Classify Digital PCR Droplets by Fitting Fluorescence Populations

Version 0.1.1.1

Description Estimates DNA target concentration by classifying digital PCR (polymerase chain reaction) droplets as positive, negative, or rain, using Expectation-Maximization Clustering. The fitting is accomplished using the 'EMMIXskew' R package (v. 1.0.3) by Kui Wang, Angus Ng, and Geoff McLachlan (2018) as based on their paper ``Multivariate Skew t Mixture Models: Applications to Fluorescence-Activated Cell Sorting Data'' <doi:10.1109/DICTA.2009.88>.

Imports mvtnorm

Depends graphics, stats, methods, grDevices, R (>= 2.10)

License GPL (>= 3)

LazyLoad yes

Encoding UTF-8

LazyData true

NeedsCompilation yes

RoxygenNote 7.1.1

Author Joyce Emlyn Guiao [aut, cre]

Maintainer Joyce Emlyn Guiao <joyce_emlyn_guiao@dlsu.edu.ph>

Repository https://cran.r-universe.dev

Date/Publication 2021-03-09 22:50:02 UTC

RemoteUrl https://github.com/cran/popPCR

RemoteRef HEAD

RemoteSha ce6087282bf94d92db743183cec1ff7692f4375f

Contents

calculateConc 2
popPCR 3
printSummaryConc 6
printSummaryFit 7
x_multiPop 8
x_onePop 8
x_twoPop 9

Index **10**

`calculateConc` *Target copies estimation*

Description

Mean target copies per partition (lambda) is derived using Poisson distribution as lambda = -ln(nneg / ntot). Target copies in sample is then calculated as conc = lambda * volSamp/(volDrp * 1000).

Usage

```
calculateConc(nneg, ntotal, volSamp, volDrp)
```

Arguments

`nneg`	numeric, negative droplet count
`ntotal`	numeric, total droplet count
`volSamp`	numeric, sample volume in microliter
`volDrp`	numeric, droplet (or partition) volume in nanoliter

Value

Returns a list with 2 named items `lambda` and `conc`

- lambda - numeric, vector of mean target copies per partition (lambda) and its lower and upper 95% confidence interval
- conc - numeric, vector of target copies in sample (based on the given sample volume (`volSamp`) and droplet volume (`volDrp`)) and its lower and upper 95% confidence interval

Examples

```
estimates <- calculateConc(5000, 20000, volSamp = 20, volDrp = 0.85)
estimates
#    Output:
#       $lambda
#          lambda     lower      upper
#       1.386294   1.362289   1.410299
```

```
#
#       $conc
#           conc      lower      upper
#       32618.69   32053.87   33183.51
```

---

popPCR *EM Mixture Model fitting of dPCR droplet fluorescence*

Description

Estimates target concentration by counting positive droplets with Poisson correction. Positive, negative, and rain populations are fitted using EM. Droplets are then classified using Maximum A Posteriori rule

Usage

```
popPCR(
  x,
  dist,
  volSamp = 20,
  volDrp = 0.85,
  maxComponents = Inf,
  negProbThres = 1e-07,
  useOnlyNegProbThres = FALSE
)
```

Arguments

x	numeric, vector of droplet fluorescence amplitude
dist	character, distribution of the mixture models ("normal", "skewed-normal", "t", "skewed-t")
volSamp	numeric, sample volume in microliter
volDrp	numeric, droplet (or partition) volume in nanoliter
maxComponents	numeric, maximum number of components (e.g. populations)
negProbThres	numeric, if only one population was detected, then its assumed as a negative population. Droplets will be classified as positive if its probability given the population < negProbThres.
useOnlyNegProbThres	logical, if TRUE, then droplets will be classified as positive if its probability given the leftmost population < negProbThres. Default is FALSE, i.e. classification is done by Maximum A Posteriori rule.

Value

Returns a `result.popPCR` S4 class object with attributes

- classification - character, vector of droplet classification
- dist - character, user-specified parameter for the mixture model
- dropletCount - list, droplet classification count
- em - list, returned value of EMMIXskew's EmSkew()
- estConc - list, estimated target concentration as lambda and sample concentration (with 95% CI)
- G - numeric, number of components fitted
- memberProb - list, component membership probability of all droplets

Examples

```
library(popPCR)

# Plot histograms of available data
hist(x_onePop, breaks = 100)
hist(x_twoPop, breaks = 100)
hist(x_multiPop, breaks = 100)

# ---- Mixture model fitting ---- #
# Example 1. One population sample
result <- popPCR(x_onePop, dist = "t")
printSummaryConc(result)
#    Output:
#        Populations detected : 1
#        Total droplets : 8000
#        Positive : 1 (0.01%)
#        Negative : 7999 (99.99%)
#
#        Target copies in sample          : 2.9414 ( 95% CI: [ -2.8237 , 8.7064 ] )
#        Mean target copies per partition : 1e-04 ( 95% CI: [ -1e-04 , 4e-04 ] )
printSummaryFit(result)
#    Output:
#        Results of fitting a 1-component t mixture model
#
#        Negative Population
#        Mix prop. : 1
#        Mu        : 1024.1614
#        Sigma     : 35253.1747
#        Dof       : 2.005

# (Option) increase negProbThres to classify negative droplets more strictly
result <- popPCR(x_onePop, dist = "t", negProbThres = 1e-4)
printSummaryConc(result)
#    Output:
#        Populations detected : 1
#        Total droplets : 8000
#        Positive : 691 (8.64%)
```

```
#        Negative : 7309 (91.36%)
#
#       Target copies in sample         : 2125.5312 ( 95% CI: [ 1966.9936 , 2284.0688 ] )
#        Mean target copies per partition : 0.0903 ( 95% CI: [ 0.0836 , 0.0971 ] )

# Example 2. Two population sample
result <- popPCR(x_twoPop, dist = "t")
printSummaryConc(result)
#    Output:
#        Populations detected : 2
#        Total droplets : 10254
#        Positive : 8693 (84.78%)
#        Negative : 1561 (15.22%)
#
#      Target copies in sample        : 44290.3819 ( 95% CI: [ 43215.6408 , 45365.1231 ] )
#        Mean target copies per partition : 1.8823 ( 95% CI: [ 1.8367 , 1.928 ] )
printSummaryFit(result)
#    Output:
#        Results of fitting a 2-component t mixture model
#
#        Negative Population
#        Mix prop. : 0.1522
#        Mu        : 2136.7435
#        Sigma     : 4126.8357
#        Dof       : 12.3562
#
#        Positive Population
#        Mix prop. : 0.8478
#        Mu        : 7580.1275
#        Sigma     : 42621.1894
#        Dof       : 2.415


# Example 3. Multiple population sample
result <- popPCR(x_multiPop, dist = "t", maxComponents = 4)
printSummaryConc(result)
#    Output:
#        Populations detected : 4
#        Total droplets : 1814
#        Positive : 44 (2.43%)
#        Negative : 1252 (69.02%)
#        Rain (1) : 258 (14.22%)
#        Rain (2) : 260 (14.33%)
#
#       Target copies in sample         : 8724.5195 ( 95% CI: [ 7999.0578 , 9449.9812 ] )
#        Mean target copies per partition : 0.3708 ( 95% CI: [ 0.34 , 0.4016 ] )

# In the output above, we see 2 rain populations! Let's examine its plot.
plot(stats::density(x_multiPop))
# We can see that Rain (1) is very close to the Negative population.
# Let's include droplets in Rain (1) in the negative droplet count.
nNegative <- result@dropletCount$neg + result@dropletCount$rain1
nTotal <- result@dropletCount$total
```

```
# Re-estimate concentration as follows
newEstimates <- calculateConc(nNegative, nTotal, volSamp = 20, volDrp = 0.85)
newEstimates
#    Output:
#       $lambda
#          lambda     lower     upper
#       0.1834247 0.1627763 0.2040731
#
#       $conc
#           conc    lower    upper
#       4315.875 3830.031 4801.719
```

`printSummaryConc` *Print result summary of popPCR*

Description

Summarizes the number of populations detected, total droplets, and number of classified positive, negative, and rain droplets. Also calculates the target copies in sample and the mean target copies per partition (lambda).

Usage

```
printSummaryConc(result.popPCR)
```

Arguments

`result.popPCR`	returned value of popPCR()

Examples

```
result <- popPCR(x_twoPop, dist = "t")
printSummaryConc(result)
#    Output:
#        Populations detected : 2
#        Total droplets : 10254
#        Positive : 8693 (84.78%)
#        Negative : 1561 (15.22%)
#
#      Target copies in sample        : 44290.3819 ( 95% CI: [ 43215.6408 , 45365.1231 ] )
#        Mean target copies per partition : 1.8823 ( 95% CI: [ 1.8367 , 1.928 ] )
```

printSummaryFit *Print fitted mixture model estimates from popPCR*

Description

Summarizes the number of populations fitted and their estimate distribution parameters. If only 1 population was detected, then it is assumed and is identified to be a negative population. If 2 populations were detected, then the leftmost is identified as the Negative Population and the rightmost is the Positive Population. If 3 or more populations were detected, then the populations between the leftmost and the rightmost will be considered as Rain Populations; which are numbered to make it identifiable in case of multiple Rain Populations (i.e. Rain (1) and Rain (2)).

Usage

```
printSummaryFit(result.popPCR)
```

Arguments

result.popPCR	returned value of popPCR()

Examples

```
result <- popPCR(x_twoPop, dist = "t")
printSummaryFit(result)
#    Output:
#        Results of fitting a 2-component t mixture model
#
#        Negative Population
#        Mix prop. : 0.1522
#        Mu        : 2136.7435
#        Sigma     : 4126.8357
#        Dof       : 12.3562
#
#        Positive Population
#        Mix prop. : 0.8478
#        Mu        : 7580.1275
#        Sigma     : 42621.1894
#        Dof       : 2.415
result <- popPCR(x_multiPop, dist = "t", maxComponents = 4)
printSummaryFit(result)
#     Output:
#        Results of fitting a 4-component t mixture model
#
#        Negative Population
#        Mix prop. : 0.6896
#        Mu        : 1452.1416
#        Sigma     : 12526.8931
#        Dof       : 21.3612
#
```

```
#        Rain (1) Population
#        Mix prop. : 0.142
#        Mu        : 2142.1118
#        Sigma     : 10762.5474
#        Dof       : 186.2947
#
#        Rain (2) Population
#        Mix prop. : 0.1457
#        Mu        : 5119.0039
#        Sigma     : 334959.2499
#        Dof       : 2.3626
#
#        Positive Population
#        Mix prop. : 0.0227
#        Mu        : 8505.9682
#        Sigma     : 192858.9044
#        Dof       : 149.8677
```

---

x_multiPop *dPCR sample w/ >=3 populations*

Description

The reaction with ID 373 in the `Dataset.zip` provided in the repository from Lievens et. al. (2017)

Usage

```
x_multiPop
```

Format

A numeric vector with 1814 droplet fluorescence

Source

https://github.com/Gromgorgel/ddPCR/blob/master/Dataset.zip

---

x_onePop *dPCR sample w/ 1 population*

Description

Simulated dataset with very high presence of rain and true mean copies per partition of 0.1.

Usage

```
x_onePop
```

Format

A numeric vector with 8000 droplet fluorescence

`x_twoPop` *dPCR sample w/ 2 populations*

Description

The reaction with ID 9 in the `Dataset.zip` provided in the repository from Lievens et. al. (2017)

Usage

```
x_twoPop
```

Format

A numeric vector with 10254 droplet fluorescence

Source

https://github.com/Gromgorgel/ddPCR/blob/master/Dataset.zip

Index

∗ **datasets**
x_multiPop, 8
x_onePop, 8
x_twoPop, 9

calculateConc, 2

popPCR, 3
printSummaryConc, 6
printSummaryFit, 7

x_multiPop, 8
x_onePop, 8
x_twoPop, 9