Package: npem (via r-universe)

May 7, 2026

Version 0.53-2

Date 2026-05-05

Title Analysis of Cell Proliferation Assays

Author Karl W Broman [aut, cre]
(<https://orcid.org/0000-0002-4914-6671>)

Maintainer Karl W Broman <broman@wisc.edu>

Description Analysis of cell proliferation assays with a normal-Poisson mixture model. Broman et al. (1996) <doi:10.1016/S0022-1759(96)00136-6>.

Depends R (>= 2.10.1)

Imports stats

License GPL-3

URL https://github.com/kbroman/npem

BugReports https://github.com/kbroman/npem/issues

LazyData true

Encoding UTF-8

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

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

Date/Publication 2026-05-07 04:04:34 UTC

RemoteUrl https://github.com/kbroman/npem

RemoteRef HEAD

RemoteSha 83c23b4ef80256f4f5b82c8b6862514064l7ce13

Contents

npem.em 2
npem.ll 4
npem.sem 5

npem.start 7
npsim 9
p711 10
p713 11
p713.2 11

Index **13**

`npem.em` *Fit the normal-Poisson model to data on a cell proliferation assay*

Description

Uses a version of the EM algorithm to fit the normal-Poisson mixture model to data on a cell proliferation assay.

Usage

```
npem.em(
  y,
  ests,
  cells = 10^6,
  n = c(24, 24, 24, 22),
  n.plates = 1,
  use.order.constraint = TRUE,
  maxit = 2000,
  tol = 0.000001,
  maxk = 20,
  prnt = 0
)
```

Arguments

`y`	Vector of transformed scintillation counts, in lexicographical order (plate by plate and group by group within a plate.)
`ests`	Initial parameter estimates, as a vector of length n.groups + 3*n.plates, of the form (λ's, (a, b, σ)'s), where λ is the average number of responding cells per 10^6 cells for a group, and (a, b, σ) are the plate-specific parameters.
`cells`	Number of cells per well. The λ's will be rescaled to give response per 10^6 cells. This may be either a single number (if all wells have the same number of cells, or 10^6 if one wishes the λ's to not be rescaled), a value for each plate (vector of length `n.plates`, or a value for each well (a vector of the same length as `y`).
`n`	Vector giving the number of wells within each group. This may have length either n.groups (if all plates have the same number of wells per group) or n.groups*n.plates.
`n.plates`	The number of plates in the data.

`use.order.constraint`
: If TRUE, force the constraint $\lambda_0 \leq \lambda_i$ for all $i \geq 1$; otherwise, no constraints are applied.

`maxit`
: Maximum number of EM iterations to perform.

`tol`
: Tolerance to determine when to stop the EM algorithm.

`maxk`
: Maximum k value in sum calculating E(k | y).

`prnt`
: If 0, don't print anything; if 1, print the log likelihood at each iteration; and if 2, print the est's and the log lik. at each iteration.

Details

Calculations are performed in a C routine. [I should describe the normal-Poisson mixture model here.]

Value

`ests`
: The estimated parameters in same form as the input argument `ests`.

`k`
: The estimated number of responding cells per well, $E(k|y)$.

`ksq`
: $E(k^2|y)$

`loglik`
: The value of the log likelihood at `ests`.

`n.iter`
: Number of iterations performed.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman et al. (1996) Estimation of antigen-responsive T cell frequencies in PBMC from human subjects. J Immunol Meth 198:119-132
Dempster et al. (1977) Maximum likelihood estimation from incomplete data via the EM algorithm. J Roy Statist Soc Ser B 39:1-38

See Also

`npem.sem()`, `npem.start()`, `npsim()`, `npem.ll()`

Examples

```
# get access to an example data set
data(p713)

# analysis of plate3
  # get starting values
start.pl3 <- npem.start(p713$counts[[3]],n=p713$n)
  # get estimates
out.pl3 <- npem.em(p713$counts[[3]],start.pl3,n=p713$n)
  # look at log likelihood at starting and ending points
```

```
npem.ll(p713$counts[[3]],start.pl3,n=p713$n)
npem.ll(p713$counts[[3]],out.pl3$ests,n=p713$n)
  # repeat with great precision, starting at previous endpoint
out.pl3 <- npem.em(p713$counts[[3]],out.pl3$ests,
                   n=p713$n,tol=1e-13)
  # run SEM algorithm to get standard errors
out.sem.pl3 <- npem.sem(p713$counts[[3]],out.pl3,n=p713$n)
round(out.pl3$ests,3)
round(out.sem.pl3$se,3)

# repeat the above for the pair, plates 3 and 4
  # get starting values
start.pl34 <- npem.start(unlist(p713$counts[3:4]),n=p713$n,n.plates=2)
  # get estimates
out.pl34 <- npem.em(unlist(p713$counts[3:4]),start.pl34,n=p713$n,n.plates=2)
  # look at log likelihood at starting and ending points
npem.ll(unlist(p713$counts[3:4]),start.pl34,n=p713$n,n.plates=2)
npem.ll(unlist(p713$counts[3:4]),out.pl34$ests,n=p713$n,n.plates=2)
  # repeat with great precision, starting at previous endpoint
out.pl34 <- npem.em(unlist(p713$counts[3:4]),out.pl34$ests,
                   n=p713$n,tol=1e-13,n.plates=2)
  # run SEM algorithm to get standard errors
out.sem.pl34 <- npem.sem(unlist(p713$counts[3:4]),out.pl34,n=p713$n,n.plates=2)
round(out.pl34$ests,3)
round(out.sem.pl34$se,3)
```

`npem.ll` *Calculate the log likelihood for the normal-Poisson model*

Description

Calculates the log likelihood at a single point in the parameter space, for the normal-Poisson mixture model with data on a cell proliferation assay.

Usage

```
npem.ll(y, ests, cells = 10^6, n = c(24, 24, 24, 22), n.plates = 1, maxk = 30)
```

Arguments

`y`	Vector of transformed scintillation counts, in lexicographical order (plate by plate and group by group within a plate.)
`ests`	Value of the parameters at which to calculate the log likelihood, as a vector of length n.groups + 3*n.plates, of the form (λ's, (a, b, σ)'s), where λ is the average number of responding cells per 10^6 cells for a group, and (a, b, σ) are the plate-specific parameters.

cells	Number of cells per well. The λ's will be rescaled to give response per 10^6 cells. This may be either a single number (if all wells have the same number of cells, or 10^6 if one wishes the λ's to not be rescaled), a value for each plate (vector of length n.plates, or a value for each well (a vector of the same length as y).
n	Vector giving the number of wells within each group. This may have length either n.groups (if all plates have the same number of wells per group) or n.groups*n.plates.
n.plates	The number of plates in the data.
maxk	Maximum k value in sum calculating $E(k\|y)$.

Details

Calculations are performed in a C routine.

Value

loglik	The log likelihood function calculated at the point ests in the parameter space.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman et al. (1996) Estimation of antigen-responsive T cell frequencies in PBMC from human subjects. J Immunol Meth 198:119-132

See Also

npem.em()

Examples

```
data(p713)
start.pl3 <- npem.start(p713$counts[[3]],n=p713$n)
out.pl3 <- npem.em(p713$counts[[3]],start.pl3,n=p713$n)
npem.ll(p713$counts[[3]],start.pl3,n=p713$n)
npem.ll(p713$counts[[3]],out.pl3$ests,n=p713$n)
```

---

npem.sem *Obtain standard errors for the estimates from npem.em*

Description

Uses the SEM algorithm to obtain estimated standard errors for the MLEs obtained after fitting the normal-Poisson mixture model to data on a cell proliferation assay.

Usage

```
npem.sem(
  y,
  npem.em.out,
  cells = 10^6,
  start = npem.em.out$ests * 1.05 + 0.001,
  n = c(24, 24, 24, 22),
  n.plates = 1,
  use.order.constraint = TRUE,
  all.se = TRUE,
  tol = 0.000001,
  maxk = 20,
  prnt = 0,
  do.var = TRUE,
  maxit = 1000
)
```

Arguments

y	Vector of transformed scintillation counts, in lexicographical order (plate by plate and group by group within a plate.)
npem.em.out	Output from the function npem.em().
cells	Number of cells per well. The λ's will be rescaled to give response per 10^6 cells. This may be either a single number (if all wells have the same number of cells, or 10^6 if one wishes the λ's to not be rescaled), a value for each plate (vector of length n.plates, or a value for each well (a vector of the same length as y).
start	Starting estimates, some small distance away from the MLE. A vector of the form (λ's, (a, b, σ)'s).
n	Vector giving the number of wells within each group. This may have length either n.groups (if all plates have the same number of wells per group) or n.groups*n.plates.
n.plates	The number of plates in the data.
use.order.constraint	If TRUE, force the constraint $\lambda_0 \leq \lambda_i$ for all $i \geq 1$; otherwise, no constraints are applied.
all.se	If TRUE, do the full SEM algorithm; if FALSE, ignore the plate-specific parameters (a,b,σ)'s, to get estimated SEs for the λ's.
tol	Tolerance to determine when to stop the EM algorithm.
maxk	Maximum k value in sum calculating $E(k\|y)$.
prnt	If 0, don't print anything; if 1, print out (at each step) the rate matrix and which elements have converged.
do.var	If TRUE, calculate the variance-covariance matrix and standard errors; if FALSE, only calculate the full-data information matrix and rate matrix.
maxit	Maximum number of iterations to perform.

Details

Calculations are performed in a C routine. It is important to first run `npem.em()` with a very small value for `tol`, such as 10^{-13}.

Value

`infor`	The full-data information matrix
`rates`	The rate matrix ("DM" in Meng and Rubin's notation).
`n.iter`	Number of iterations performed in calculating the rate matrix.
`var`	The estimated variance-covariance matrix.
`se`	The estimated standard errors. (The square root of the diagnol of `var`.)

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman et al. (1996) Estimation of antigen-responsive T cell frequencies in PBMC from human subjects. J Immunol Meth 198:119-132
Dempster et al. (1977) Maximum likelihood estimation from incomplete data via the EM algorithm. J Roy Statist Soc Ser B 39:1-38
Meng and Rubin (1991) Using EM to obtain asymptotic variance-covariance matrices: the SEM algorithm. J Am Statist Asso 86:899-909

See Also

`npem.em()`

Examples

```
data(p713)
start.pl3 <- npem.start(p713$counts[[3]],n=p713$n)
out.pl3 <- npem.em(p713$counts[[3]],start.pl3,n=p713$n,tol=1e-13)
out.sem.pl3 <- npem.sem(p713$counts[[3]],out.pl3,n=p713$n)
```

---

npem.start *Obtain approximate starting values for npem.em*

Description

Obtains crude starting values, needed for the EM algorithm.

Usage

```
npem.start(
  y,
  cells = 10^6,
  n = c(24, 24, 24, 22),
  n.plates = 1,
  n.groups = 4,
  n.sd = 2,
  cv = 0.33
)
```

Arguments

y	Vector of transformed scintillation counts, in lexicographical order (plate by plate and group by group within a plate.)
cells	Number of cells per well. The λ's will be rescaled to give response per 10^6 cells. This may be either a single number (if all wells have the same number of cells, or 10^6 if one wishes the λ's to not be rescaled), a value for each plate (vector of length n.plates, or a value for each well (a vector of the same length as y).
n	Vector giving the number of wells within each group. This may have length either n.groups (if all plates have the same number of wells per group) or n.groups*n.plates.
n.plates	The number of plates in the data.
n.groups	The number of groups. (This is needed here but not elsewhere, because usually I figure it out from n.plates and the length of the argument ests.)
n.sd	Number of SDs above the mean to use as a cutoff
cv	Coefficient of variation (= SD/ave) used in randomizing the starting point; use cv=0 to avoid randomization.

Details

[I should describe the algorithm in more detail here.]

Value

ests	The parameter estimates to use as starting values for the EM algorithm, as a vector of length n.groups + 3*n.plates, of the form (λ's, (a, b, σ)'s), where λ is the average number of responding cells per 10^6 cells for a group, and (a, b, σ) are the plate-specific parameters.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman et al. (1996) Estimation of antigen-responsive T cell frequencies in PBMC from human subjects. J Immunol Meth 198:119-132

See Also

npem.em()

Examples

```
data(p713)
start.pl3 <- npem.start(p713$counts[[3]],n=p713$n)
out.pl3 <- npem.em(p713$counts[[3]],start.pl3,n=p713$n,tol=1e-13)
```

---

npsim *Simulate data for the normal-Poisson mixture model*

Description

Simulate data for a single plate using the normal-Poisson mixture model.

Usage

```
npsim(ests, n = c(24, 24, 24, 22))
```

Arguments

ests	Parameter values as a vector for the form (λ_1,...,λ_n), a, b, σ
n	Lengths of the λ groups

Value

k	The simulated numbers of responding cells per well, useful for comparison with those estimated from npem.em().
y	The simulated transformed scintillation counts, to be used as input for npem.em().

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman et al. (1996) Estimation of antigen-responsive T cell frequencies in PBMC from human subjects. J Immunol Meth 198:119-132

See Also

npem.em()

Examples

```
ests <- c(0.5,1.5,2.5,5,15,5)
n <- c(24,24,22)
dat <- npsim(ests,n)
out <- npem.em(dat$y,ests=ests,n=n)
jitter <- runif(length(dat$k),-0.1,0.1)
plot(dat$k + jitter, out$k, xlab="True no. cells",
     ylab="Estimated no. cells", lwd=2)
plot(dat$y,out$k,type="n",xlab="Response",ylab="Estimated no. cells")
for(i in unique(dat$k))
  points(dat$y[dat$k==i],out$k[dat$k==i],col=i+1,lwd=2)
```

---

p711 *Data for a limiting dilution assay*

Description

This is data for a limiting dilution assay for a single patient, composed of 6 pairs of plates at 6 different cell concentrations. The data is from the same subject (at the same time) as p713.2().

Usage

```
p711
```

Format

The data is a list with three components:

counts	A list of length 12, each component of which is a vector giving the square-root-transformed scintillation counts for a
cells	A vector of length 12, giving the estimated number of cells per well for each of the 12 plates.
n	Vector of length 4, giving the number of wells per group, which is the same for each plate.

Author(s)

Karl W Broman, <broman@wisc.edu>
https://github.com/kbroman/npem

Source

Michael Tigges, Chiron Biocine

See Also

p713.2(), p711(), npem.em()

p713 *Data for a limiting dilution assay*

Description

This is data for a limiting dilution assay for a single patient, composed of 6 pairs of plates at 6 different cell concentrations.

Usage

```
p713
```

Format

The data is a list with three components:

counts	A list of length 12, each component of which is a vector giving the square-root-transformed scintillation counts for a
cells	A vector of length 12, giving the estimated number of cells per well for each of the 12 plates.
n	Vector of length 4, giving the number of wells per group, which is the same for each plate.

Author(s)

Karl W Broman, <broman@wisc.edu>
https://github.com/kbroman/npem

Source

Michael Tigges, Chiron Biocine

See Also

p713(), p713.2(), npem.em()

p713.2 *Data for a limiting dilution assay*

Description

This is data for a limiting dilution assay for a single patient, composed of 5 pairs of plates at 5 different cell concentrations. The data is from the same subject (at the same time) as p713().

Usage

```
p713.2
```

Format

The data is a list with three components:

counts	A list of length 10, each component of which is a vector giving the square-root-transformed scintillation counts for a
cells	A vector of length 10, giving the estimated number of cells per well for each of the 10 plates.
n	Vector of length 4, giving the number of wells per group, which is the same for each plate.

Author(s)

Karl W Broman, <broman@wisc.edu>
https://github.com/kbroman/npem

Source

Michael Tigges, Chiron Biocine

See Also

p713(), p711(), npem.em()

Index

∗ **datagen**
npsim, 9
∗ **datasets**
p711, 10
p713, 11
p713.2, 11
∗ **models**
npem.em, 2
npem.ll, 4
npem.sem, 5
∗ **utilities**
npem.start, 7

npem.em, 2
npem.em(), *5–7*, *9–12*
npem.ll, 4
npem.ll(), *3*
npem.sem, 5
npem.sem(), *3*
npem.start, 7
npem.start(), *3*
npsim, 9
npsim(), *3*

p711, 10
p711(), *10*, *12*
p713, 11
p713(), *11, 12*
p713.2, 11
p713.2(), *10, 11*