Noncompartmental Analysis

Kyun-Seop Bae MD PhD

2022-10-25

Contents

1 Using NonCompart::sNCA

1.1 Introduction

Noncompartmental Analysis(NCA) is relatively robust and used frequently for regulatory purpose.

1.2 Individual NCA parameters

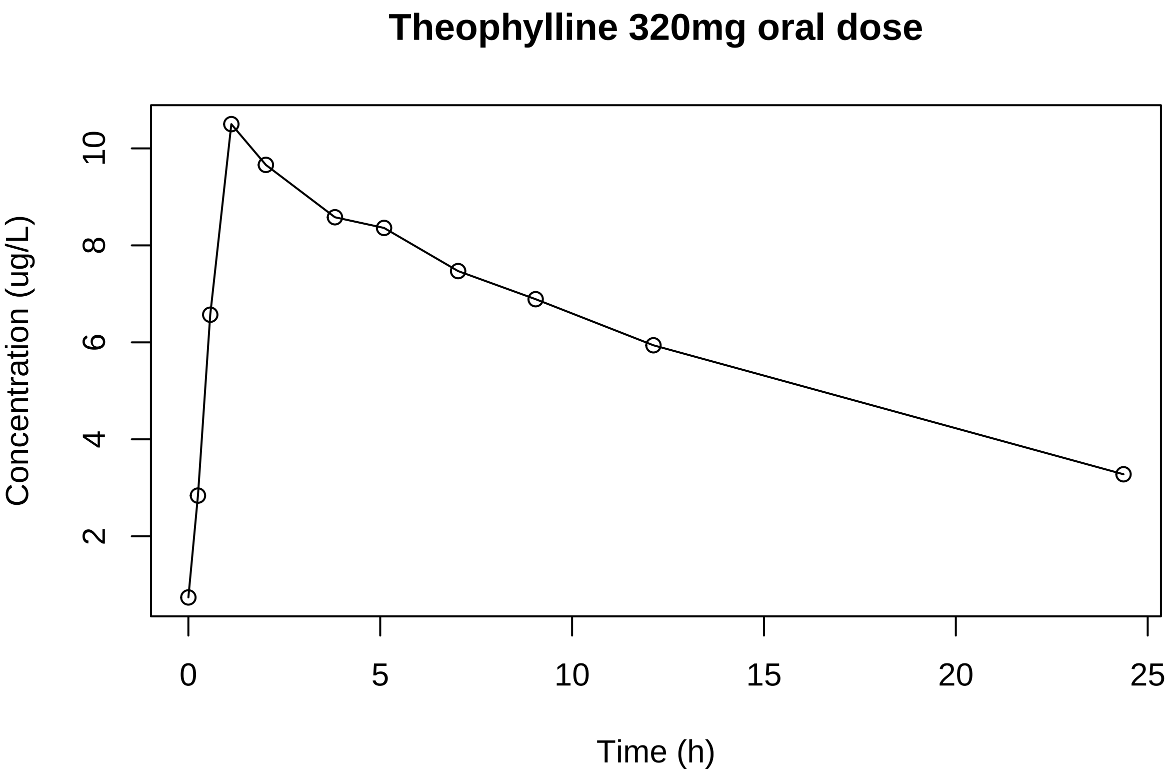
1.2.1 Preparation

This is just for the prepartion of data for the subsequent R scripts.

```
Adm = c("BOLUS", "INFUSION", "EXTRAVASCULAR")[3] # Drug Administration Method
Dose = 320 # mg
x = x0 = Theoph[Theoph$Subject==1, "Time"] # h
y = y0 = Theoph[Theoph$Subject==1, "conc"] # ug/L

# For the calculation of AUClast
iLastNonZero = max(which(y > 0)) # index of last non-zero concentration
x1 = x0[1:iLastNonZero]
y1 = y0[1:iLastNonZero]

# For the log-concentration vs. time regression
x2 = x0[y0 > 0]
y2 = y0[y0 > 0]

# Print data
cbind(Time=x0, Conc=y0)
```

```
      Time  Conc
 [1,] 0.00  0.74
 [2,] 0.25  2.84
 [3,] 0.57  6.57
 [4,] 1.12 10.50
 [5,] 2.02  9.66
 [6,] 3.82  8.58
 [7,] 5.10  8.36
 [8,] 7.03  7.47
 [9,]  9.05  6.89
[10,] 12.12  5.94
[11,] 24.37  3.28
```

1.2.2 Plot of raw data



1.2.3 Cmax (CMAX)

Maximum concentration.

```
CMAX = max(y)
CMAX
```

```
[1] 10.5
```

1.2.4 Cmax_D (CMAXD)

Dose normalized Cmax.

```
CMAXD = NA
if (Dose > 0) CMAXD = CMAX/Dose
CMAXD
```

```
[1] 0.0328125
```

1.2.5 Tmax (TMAX)

Time of maximum concentration.

```
TMAX = NA
if (CMAX > 0) TMAX = x[which.max(y)]
TMAX
```

```
[1] 1.12
```

1.2.6 Tlag (TLAG)

Time until first non-zero concentration.

```
TLAG = NA
if (CMAX > 0) TLAG = x[max(1, min(which(y > 0)) - 1)]
TLAG
```

```
[1] 0
```

1.2.7 Clast (CLST)

Last non-zero concentration.

```
CLST = y[iLastNonZero]
CLST
```

```
[1] 3.28
```

1.2.8 Tlast (TLST)

Time of last non-zero concentration.

```
TLST = x[iLastNonZero]
TLST
```

```
[1] 24.37
```

1.2.9 Rsq (R2)

R-squared value from the log concentration and time regression.

1.2.10 Rsq_adjusted (R2ADJ)

Adujsted R-squared value.

$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-2}$

1.2.11 Corr_XY (CORRXY)

Correlation value of the regression

1.2.12 b0 (b0)

Intercept of the simple linear regression of log(y) vs x

1.2.13 Lambda_z (LAMZ)

Terminal slope as a positive value

1.2.14 No_points_Lambda_z (LAMZNPT)

Number of points used for the regression

1.2.15 Lambda_z_lower (LAMZLL)

First time point used for the regression

1.2.16 Lambda_z_upper (LAMZUL)

Last time point used for the regression

1.2.17 Calculation of R2, R2ADJ, CORRXY, b0, LAMZ, LAMZNPT, LAMZLL, and LAMZUL

Only positive concentrations are used. In case of oral administration, the first possible point is next to Tmax point. In case of intravascular administration, the first point can be Tmax point.

If the difference of R2ADJ (R2-squared adjusted) is less than 1e-4, the longer slope is chosen. Regression points should be at least 3.

```
x = x2
y = y2

if (toupper(Adm) == "EXTRAVASCULAR") {
  iFirst = which.max(y) + 1 # for oral administration
} else {
  iFirst = which.max(y)     # for intravenous administration
}
iLast = iLastNonZero

ColNames = c("R2", "R2ADJ", "CORRXY", "b0", "LAMZ", "LAMZNPT", "LAMZLL", "LAMZUL")
mRes = matrix(nrow = iLast - iFirst + 1 - 2, ncol=length(ColNames))
colnames(mRes) = ColNames
for (i in iFirst:(iLast - 2)) {
  Res = lm(log(y[i:iLast]) ~ x[i:iLast])
  mRes[i - iFirst + 1, "R2"]      = summary(Res)$r.squared
  mRes[i - iFirst + 1, "R2ADJ"]   = summary(Res)$adj.r.squared
  mRes[i - iFirst + 1, "CORRXY"]  = cor(log(y[i:iLast]), x[i:iLast])
  mRes[i - iFirst + 1, "b0"]      =  Res$coefficients[1]
  mRes[i - iFirst + 1, "LAMZ"]    = -Res$coefficients[2]
  mRes[i - iFirst + 1, "LAMZNPT"] = iLast - i + 1
  mRes[i - iFirst + 1, "LAMZLL"]  = x[i]
  mRes[i - iFirst + 1, "LAMZUL"]  = x[iLast]
}
mRes
```

```
            R2     R2ADJ     CORRXY       b0       LAMZ LAMZNPT LAMZLL LAMZUL
[1,] 0.9988013 0.9985615 -0.9994005 2.355187 0.04778625       7   2.02  24.37
[2,] 0.9987305 0.9984131 -0.9993650 2.350845 0.04751440       6   3.82  24.37
[3,] 0.9995671 0.9994229 -0.9997836 2.362429 0.04817356       5   5.10  24.37
[4,] 0.9996109 0.9994164 -0.9998054 2.356834 0.04787556       4   7.03  24.37
[5,] 0.9999997 0.9999995 -0.9999999 2.368785 0.04845700       3   9.05  24.37
```

```
OKs = abs(max(mRes[,"R2ADJ"]) - mRes[,"R2ADJ"]) < 1e-4
resNCA = as.data.frame(mRes[which(OKs)[1],,drop=FALSE])
resNCA
```

```
         R2     R2ADJ     CORRXY       b0      LAMZ LAMZNPT LAMZLL LAMZUL
1 0.9999997 0.9999995 -0.9999999 2.368785 0.048457       3   9.05  24.37
```

```
attach(resNCA, warn.conflicts=FALSE)
```

If you want to manually omit some points, use an R package for convenience.

1.2.18 HL_Lambda_z (LAMZHL)

Terminal half-life calculated by ln(2)/LAMZ

```
LAMZHL = NA
if (LAMZ > 0) LAMZHL = log(2)/LAMZ
LAMZHL
```

```
[1] 14.30438
```

1.2.19 Clast_pred (CLSTP)

Predicted Clast, predicted concentration at Tlast.

$C_{last,pred} = exp(\beta_0 - \lambda \cdot T_{last})$

```
CLSTP = NA
if (LAMZ > 0) CLSTP = exp(b0 - LAMZ*x[iLast])
CLSTP
```

```
[1] 3.280146
```

1.2.20 C0 (C0)

Concentration at time 0, intial concentration. This is calculated only BOLUS administration.

$C_0 = exp(log(c_1) - t_1 \frac{log(c_2)-log(c_1)}{t_2-t_1})$

```
x = x0
y = y0
if (toupper(Adm) == "BOLUS") {
  if (y[1] > y[2] & y[2] > 0) {
    C0 = exp(log(y[1]) - x[1]*(log(y[2]) - log(y[1]))/(x[2] - x[1]))
  } else {
    C0 = y[x==min(x[y > 0])]
  }
} else {
  C0 = NA
}
```

1.2.21 AUClast (AUCLST)

Area under the time-concentration curve from dosing to the last positive concentration.

For linear trapezoidal method,

$$AUC_{last} = \sum_{i=2} \frac{(t_i - t_{i-1}) \times (c_i - c_{i-1})}{2}$$

```
n = length(x1)
AUCLST = sum((y1[-1] + y1[-n]) * (x1[-1] - x1[-n]))/2
AUCLST
```

```
[1] 148.923
```

For 'linear-up log-down' method,

```
AUCLST = 0
for (i in 2:n) {
  if (y1[i] < y1[i-1] & y1[i] > 0) {
    k = (log(y1[i - 1]) - log(y1[i]))/(x1[i] - x1[i - 1]) # slope in log
    AUCLST = AUCLST + (y1[i - 1] - y1[i])/k
  } else {
    AUCLST = AUCLST + (x1[i] - x1[i - 1])*(y1[i] + y1[i - 1])/2
  }
}
AUCLST
```

1.2.22 AUCall (AUCALL)

AUC values including all zero concentrations.

For linear trapezoidal method,

```
AUCALL = sum((y0[-1] + y0[-n]) * (x0[-1] - x0[-n]))/2
AUCALL
```

```
[1] 148.923
```

For 'linear-up log-down' method,

```
AUCALL = 0
for (i in 2:n) {
  if (y0[i] < y0[i-1] & y0[i] > 0) {
    k = (log(y0[i - 1]) - log(y0[i]))/(x0[i] - x0[i - 1]) # slope in log
    AUCALL = AUCALL + (y0[i - 1] - y0[i])/k
  } else {
    AUCALL = AUCALL + (x0[i] - x0[i - 1])*(y0[i] + y0[i - 1])/2
  }
}
AUCALL
```

Zero concentrations to be log-transformed need not be removed, because R can handle infinity value.

1.2.23 AUCinf_obs (AUCIFO)

AUCinf observed.

$$AUC_{inf,obs} = AUC_{last} + \frac{C_{last}}{\lambda_z}$$

1.2.24 AUC_%Extrap_obs (AUCPEO)

AUC percent extrapolated observed.

$$AUC_{\%Extrap,obs} = (1 - \frac{AUC_{last}}{AUC_{inf,obs}}) \times 100$$

1.2.25 AUCinf_D_obs (AUCIFOD)

Dose normalized AUCinf observed.

$$AUC_{dose,inf,obs} = fracAUC_{inf,obs}Dose$$

1.2.26 AUCinf_pred (AUCIFP)

AUCinf predicted.

$$AUC_{inf,pred} = AUC_{last} + \frac{C_{last,pred}}{\lambda_z}$$

1.2.27 AUC_%Extrap_pred (AUCPEP)

AUC percent extrapolated predicted.

$$AUC_{\%Extrap,pred} = (1 - \frac{AUC_{last}}{AUC_{inf,pred}}) \times 100$$

1.2.28 AUCinf_D_pred (AUCIFPD)

Dose normalized AUCinf predicted.

$$AUC_{dose,inf,pred} = fracAUC_{inf,pred}Dose$$

1.2.29 AUMClast (AUMCLST)

$$AUMC_{last} = \int_0^{t_{last}} tC(t)dt$$

For linear trapezoidal method;

$$AUMC_{last} \approx \sum_{i=2} \frac{(t_i - t_{i-1})(t_i c_i + t_{i-1} c_{i-1})}{2}$$

```
AUMCLST = sum((x1[-1] - x1[-n])*(x1[-1]*y1[-1] + x1[-n]*y1[-n]))/2
```

For ‘linear-up log-down’ method;

```
AUMCLST = 0
for (i in 2:n) {
  if (y1[i] < y1[i-1] & y1[i] > 0) {
    k = (log(y1[i-1]) - log(y1[i]))/(x1[i] - x1[i-1]) # slope in log
    AUMCLST = AUMCLST + (x1[i-1]*y1[i-1] - x1[i]*y1[i])/k + (y1[i-1] - y1[i])/k/k
  } else {
    AUMCLST = AUMCLST + (x1[i] - x1[i-1])*(x1[i]*y1[i] + x1[i-1]*y1[i-1])/2
  }
}
```

1.2.30 AUMCinf_obs (AUMCIFO)

AUMC infinity observed.

$$AUMC_{inf,obs} = AUMC_{last} + \frac{C_{last}T_{last}}{\lambda_z} + \frac{C_{last}}{(\lambda_z)^2}$$

1.2.31 AUMC_%Extrap_obs (AUMCPEO)

AUMC percent extrapolated observed.

$$AUMC_{\%Extrap,obs} = (1 - \frac{AUMC_{last}}{AUMC_{inf,obs}}) \times 100$$

1.2.32 AUMCinf_pred (AUMCIFP)

AUMC infinity predicted.

$$AUMC_{inf,pred} = AUMC_{last} + \frac{C_{last,pred}T_{last}}{\lambda_z} + \frac{C_{last,pred}}{(\lambda_z)^2}$$

1.2.33 AUMC_%Extrap_pred (AUMCPEP)

AUMC percent extrapolated predicted.

$$AUMC_{\%Extrap,pred} = (1 - \frac{AUMC_{last}}{AUMC_{inf,pred}}) \times 100$$

1.2.34 AUC_Back_Ext_obs (AUCBEO)

AUC back extrapolated observed. This is only for BOLUS administration.

For trapezoidal method;

$$AUC_{backextrap} = \frac{t_1 \times (C_0 + C_1)}{2}$$

For log-down method;

$$AUC_{backextrap} = \frac{t_1 \times (C_0 + C_1)}{log(C_0) - log(C_1)}$$

1.2.35 AUC_%Back_Ext_obs (AUCPBEO)

AUC percent back extrapolated observed. This is only for BOLUS administration.

$$AUC_{\%backextrap,obs} = \frac{AUC_{back_{e}xtrap}}{AUC_{inf,obs}} \times 100$$

1.2.36 AUC_%Back_Ext_pred (AUCPBEP)

AUC percent back extrapolated predicted. This is only for BOLUS administration.

$$AUC_{\%backextrap,pred} = \frac{AUC_{back_{e}xtrap}}{AUC_{inf,pred}} \times 100$$

1.2.37 MRTlast (MRTEVLST, MRTIVLST)

Mean Residence Time (MRT) from 0 to Tlast.

$$MRT_{EV,last} = \frac{AUMC_{last}}{AUC_{last}}$$

$$MRT_{IV,last} = \frac{AUMC_{last}}{AUC_{last}} - \frac{Dur}{2}$$

Here 'Dur' is infusion duration.

1.2.38 MRTinf_obs (MRTEVIFO, MRTIVIFO)

MRT infinity observed.

$$MRT_{EV,inf,obs} = \frac{AUMC_{inf,obs}}{AUC_{inf,obs}}$$

$$MRT_{IV,inf,obs} = \frac{AUMC_{inf,obs}}{AUC_{inf,obs}} - \frac{Dur}{2}$$

1.2.39 MRTinf_pred (MRTEVIFP, MRTIVIFP)

MRT infinity predicted.

$$MRT_{EV,inf,pred} = \frac{AUMC_{inf,pred}}{AUC_{inf,pred}}$$

$$MRT_{IV,inf,pred} = \frac{AUMC_{inf,pred}}{AUC_{inf,pred}} - \frac{Dur}{2}$$

1.2.40 Vz_obs (VZO) or Vz_F_obs (VZFO)

Volume of distribution by terminal slope observed. VZO is for intravascular administration and VZFO for extravascular administration.

$$V_{z,obs} = \frac{Dose}{AUC_{inf,obs} \times \lambda_z}$$

1.2.41 Vz_pred (VZP) or Vz_F_pred (VZFP)

Volume of distribution by terminal slope predicted. VZP is for intravascular administration and VZFP for extravascular administration.

$V_{z,pred} = \frac{Dose}{AUC_{inf,pred} \times \lambda_z}$

1.2.42 CL_obs (CLO) or CL_F_obs (CLFO)

Clearance observed. CLO is for intravascular administration and CLFO for extravascular administration.

$CL_{obs} = \frac{Dose}{AUC_{inf,obs}}$

1.2.43 CL_pred (CLP) or CL_F_pred (CLFP)

Clearance predicted. CLP is for intravascular administration and CLFP for extravascular administration.

$CL_{pred} = \frac{Dose}{AUC_{inf,pred}}$

1.2.44 Vss_obs (VSSO)

Volume of distribution at steady state, observed. This is for intravascular administration only.

$V_{ss,obs} = MRT_{IV,inf,obs} \times CL_{obs}$

1.2.45 Vss_pred (VSSP)

Volume of distribution at stead state, predicted. This is for intravascular administration only.

$V_{ss,pred} = MRT_{IV,inf,pred} \times CL_{pred}$

1.2.46 AmtRecLast (RCAMLST)

Amount recovered in urine.

$Ae_{last} = \sum_{i=1} Vol_i Conc_i$

1.2.47 CLrenal (RENALCL)

Renal clearance.

$CL_R = \frac{Ae_{last}}{AUC_{last}}$

1.2.48 fe (FE)

Fraction excreted unchanged.

$$fe = \frac{CL_R}{CL_{obs}} = \frac{Ae_{last}/AUC_{last}}{Dose/AUC_{inf,obs}} = \frac{Ae_{last}}{Dose} \times \frac{AUC_{inf,obs}}{AUC_{last}}$$

1.2.49 AI (ARCMAX, ARAUC, ARCMIN)

Accumulation index, accumulation ratio.

$$R_{ac} = \frac{C_{max,ss}}{C_{max,1}} = \frac{AUC_{tau,ss}}{AUC_{tau,1}} = \frac{AUC_{inf,ss}}{AUC_{inf,1}} = \frac{C_{min,ss}}{C_{min,1}}$$