Next generation modelling language

Michael R. Dunlavey
Pharsight® Corporation

Objectives

- ✓ Support both modelling & trial simulation
- ✓ Free-form, S-PLUS®-compatible syntax
- ✓ Arbitrary variable names
- ✓ C-compiled, not Fortran

Features

Modelling:

- ✓ Easy NONMEM® transition
- ✓ Multiple responses
- ✓ BQL
- ✓ Categorical & hazard-based responses
- ✓ User-defined log-likelihood
- ✓ Multiple dose routes & steady-state
- ✓ Differential equations
- ✓ Explicit closed-form
- ✓ Covariate interpolation
- ✓ Time-discontinuous models
- ✓ Unit conversion
- ✓ Non-numeric data
- ✓ Covariate selection
- ✓ Bootstrap
- ✓ Simulation & table generation
- ✓ Can call external user code

Trial simulation:

- ✓ Direct use of fitted models
- ✓ Uncertainty via MVN & Wishart distributions
- ✓ General protocol-definition syntax

Algorithms

Prototype model fitting engine:

- ✓ FO, FOCE, Laplacian, Adaptive Gaussian Quadrature
- ✓ Nonparametric
- ✓ MPI-enabled for parallel execution

ODE:

- ✓ Non-stiff, stiff, and matrix exponent
- ✓ Future: closed form for constant Jacobian models
- ✓ Automatically selected, or by user

Feature highlights

Infusion

```
# in coldef file, specify rate column
dose( a <- AMT, RATE)
```

Tlag, and zero-order dosing

```
# in model file, specify options on dosepoint
dosepoint( a, tlag = TLAG, duration = DUR)
```

Steady-state

```
# in coldef file, give SS column, and define it
ss(SS, amt bolus(a) 24 dt)
```

Time-discontinuous (e.g. reflux) models

```
# in model file, use sequence statement
sequence {
    while(1){    # loop "forever"
        sleep( t0 )        # delay by time t0
        bReflux = 1        # turn reflux on
        sleep( t1 )        # delay by time t1
        bReflux = 0        # turn reflux off
    } # and repeat
}
```

Categorical models

```
# in model file
stparm(XX1 = tvX1 + nX1,  XX2 = XX1 + tvX12)
multi(Y, ilogit, C - XX1, C - XX2)
# in coldef file, specify observation column
obs(Y <- E)
```

Hazard-based models

```
# in model file, compute hazard
stparm(haz = exp( (tvlHaz + nlHaz) + dlHazdC * C ))
# specify censor variable to observe, or count
# hazard is automatically integrated
event(cens, haz) # use this for time-to-event model
count(cnt, haz)  # use this for count model
```

Non-numeric data

```
# in coldef file, specify gender column
covr(Gender <- sex(M=0, F=1)) # and values
```

Use within S-PLUS

```
oneCpt <- ModelFragment({
    deriv( a = - a * ke )
    dosepoint(a)
    c = a / v
})
mixMod01 <- ModelFragment({
    fixef(
        lke0 = c(-6.9, -2.3, 0)
        , lv0 = c(0, 2.3, 6.9)
        )
    ranef(
        diag(nlke, nlv) = c(1,1)
        )
    covariate(Wt)
    stparm(
        ke = exp(lke0 + nlke)
        , v = exp(lv0 + nlv)
        )
})
errMod01 <- ModelFragment({
    error(eps0)
    observe(cObs = c + eps0)
})
ModelGen("temp"
    , c(oneCpt, mixMod01, errMod01))
```

Results

Performance and results are comparable to NONMEM and S-PLUS NLME on a large suite of models.
Language expresses both NLME and trial simulation models.

Trademarks

S-PLUS, Insightful, Inc.
NONMEM, Globomax, Inc.

Contact

Michael Dunlavey, PhD
Pharsight Corporation
Ph: +1 781.449.2719
Email: mdunlavey@pharsight.com

NONMEM example

```
$PROBLEM PHENOBARB  SIMPLE MODEL
 $INPUT    ID TIME AMT WGT APGR DV
 $DATA     pheno.dat IGNORE=#
 $SUBR    ADVAN1
 $PK
       TVCL=THETA(1)*WGT
       TVV=THETA(2)*WGT
       CL=TVCL*EXP(ETA(1))
       V=TVV*EXP(ETA(2))
       K=CL/V
       S1=V
 $THETA     (0,.005)   (0, 1)
 $OMEGA  .25    .25
 $ERROR
 Y=F+EPS(1)
 $SIGMA 8
 $ESTIMATION METHOD=1   POSTHOC MAXEVALS=9999
 $COV
```

New language model file

```
Pheno(){                          # model name
    dosepoint( a )                # indicate variable to receive dose
    covariate( wt )               # declare that wt is a covariate
    deriv( a =  - a * Cl/V )      # differential equation for central cpt
    c = a / V                     # define concentration variable
    fixef( wtCl = c(,.005,), wtV = c(,0.9,)) # define fixed effects
    ranef( diag(nCl, nV) = c(0.04, 0.04))    # define random effects
    stparm( Cl = (wtCl * wt) * exp(nCl))     # define structural parameter Cl
    stparm( V = (wtV * wt) * exp(nV))        # define structural parameter V
    error(eps1 = 10)                         # define error variable w. initial std deviation
    observe( cObs = c + eps1 )               # define observation variable
}
```

Column definition file

```
id( ID )             # which column specifies individual id
dose( a <- AMT )     # which column gives dose for dosepoint a
obs( cObs <- CONC )  # which column gives observed concentration
time( T )            # which column gives time of dose or observation
covr( wt <- WT)      # which column gives weight covariate
```