Modelling Description Language

Nick Holford and Mike K Smith

Drug Disease Model Resources

"Builds and maintains a universally applicable, open source, model based framework, intended as the gold standard for future collaborative drug and disease Modelling & Simulation"

http://www.ddmore.eu/content/project

MDL Structure
Model Coding Language – Model Object 1

Evolutionary step from NM-TRAN

Model Coding Language – Model Object 2

Evolutionary step from NM-TRAN

Model Coding Language – Model Object 3

Evolutionary step from NM-TRAN
Model Coding Language – Data Object

```r
FILE:
data=list(source="warfarin_conc_pca.csv", 
  ignore="#", 
  inputformat="NONMEM")

HEADER:
  ID=list(type=categorical)
  TIME=list(type=continuous)
  WT=list(type=continuous, units="kg")
  AGE=list(type=continuous, units="")
  SEX=list(type=categorical(female=1,male=0))
  AMT=list(type=continuous)
  DVID=list(type=categorical)
  DV=list(type=continuous)
  MDV=list(type=categorical)
}
```

Model Coding Language – Parameter Object

```r
POPULATION:
  POP_CL=list(value=0.1,lo=0.001)
  POP_V=list(value=8,lo=0.001)
  POP_KA=list(value=2,lo=0.001)
  POP_TLAG=list(value=1,lo=0.001)

VARIABILITY:
  PPV_CL=0.1,
  PPV_V=0.1

VARY:
  matrix(PPV_CL=0.1, PPV_V=0.1)

THETA:
  (0.001,0.1) \; \text{POP}\_CL \; \text{L/h/70kg}
  (0.001,8) \; \text{POP}\_V \; \text{L/70kg}
  (0.001,2) \; \text{POP}\_KA \; \text{h-1}
  (0.001,1) \; \text{POP}\_TLAG \; \text{h}

OMEGA:
  BLOCK(2)
  0.1 \; \text{PPV}\_CL
  0.01 \; \text{PPV}\_V

OMEGA BLOCK:
  0.1 \; \text{PPV}\_KA
  0.1 \; \text{PPV}\_TLAG

SIGMA:
  0.01 \; \text{RUV}\_PROP
  0.05 \; \text{RUV}\_ADD \; \text{mg/L}
```

Model Coding Language – Task Properties Object

```r
DATA: IGNORE (DVID.EQ.2) ; ignore PCA observations
```

```r
ESTIMATE:
  target=t
  model=m
  parameter=p
  data=d
  algo=list("COND  INTER")
  max=9990
  sig=3
  cov="y"
```

```r
THETA:
  (0.001,0.1) \; \text{POP}\_CL \; \text{L/h/70kg}
  (0.001,8) \; \text{POP}\_V \; \text{L/70kg}
  (0.001,2) \; \text{POP}\_KA \; \text{h-1}
  (0.001,1) \; \text{POP}\_TLAG \; \text{h}

OMEGA BLOCK:
  0.1 \; \text{PPV}\_CL
  0.01 \; \text{PPV}\_V

OMEGA:
  0.1 \; \text{PPV}\_KA
  0.1 \; \text{PPV}\_TLAG

SIGMA:
  0.01 \; \text{RUV}\_PROP
  0.05 \; \text{RUV}\_ADD \; \text{mg/L}
```
Task Execution Language

- MCL = Nouns, TEL = Verbs, MCL Task Properties = Adverbs.
  
  GET <<Model + Data + Parameter initial values>> and
  DO <<Estimation>>
  (LIKE THIS <<Task Properties>>)  

- Tasks define what MCL objects are required:  
  • Estimation = Model + Data + Parameters (initial, bounds) + Task Properties (Settings)  
  • Simulation / Optimal Design = Model + Data Design + Parameters (point estimates or distributions) + Task Properties (Settings)

Evolutionary step from NM-TRAN

Model Coding Language – Task Properties Object

```
warfarin_PK_CONC_task = taskobj({
  DATA{IGNORE=if(DVID==2)}
  myEST=function(t,m,p,d) {
    ESTIMATE{
      target=t
      model=m
      parameter=p
      data=d
      algorithm="COND INTER"
      max=9990
      sig=3
      cov="y"
    }
  }
})
```

- DATA IGNORE (DVID.EQ.2) ; ignore PCA observations  
- EST METHOD=COND INTER MAX=9990 SIG=3 COV  

Evolutionary step from WFN, PsN, etc.

Task Execution Language – TEL Command Object

```
warfarin_PK_CONC_tel = telobj(
  myEST(t="NONMEM", NONMEM WFN Matlab BUGS R)
)
```

- TEL defines basic tasks that can build to more complex workflows.
Use R for general data and statistical tasks

Task Execution Language – R Command Object

Your favourite R script  Your favourite R script

Translation to other languages

"Rosetta Stone"

MDL (alternative random effects)

 MCL

MLTRAN

BUGS

Phoenix Modeling Language (PML)

GROUP_VARIABLES

\{ 

GRPV = POP_V * WT/70 

... 

\}

RANDOM_VARIABLE_DEFINITION

\{ 

lnV ~ (type=Normal, mean=log(GRPV), variance=PPV_V, level=ID) 

... 

\}

INDIVIDUAL_VARIABLES

\{ 

V = exp(lnV) 

... 

\}

EQUATION:

\{ 

GRPV = POP_V * (WT/70) 

... 

\}

DEFINITION:

\{ 

V = \{distribution=logNormal, prediction=GRPV, standardDeviation=PPV_V\} 

... 

\}

MLTRAN

MODEL

\{ 

GROUP

\{ 

GRPV = POP_V * (WT/70) 

... 

\}

DEFINITION:

\{ 

V = \{distribution=logNormal, prediction=GRPV, standardDeviation=PPV_V\} 

... 

\}

EQUATION:

\{ 

GRPV = POP_V * (WT/70) 

... 

\}

DEFINITION:

\{ 

V = \{distribution=logNormal, prediction=GRPV, standardDeviation=PPV_V\} 

... 

\}

BUGS

model{ 

for (i in 1:N) { 

lnV[i] ~ dnorm(LOG.GRPV,PPV_V) 

... 

V[i] = exp(lnV[i]) 

... 

}

}

Phoenix Modeling Language (PML)

MDL and the rest of DDMoRe
MDL and the rest of DDMoRe

What else?

Model Coding Language
- Modular models combining library functions
- Levels of random effect
- Non-normal distributions
- "odd type data" statements
  - POISSON
  - CATEGORICAL
  - HAZARD

Task Execution Language
- Target software appropriate to task using the same model
- Mix and match using library modelling object groups
- Workflow of tasks through framework

Active engagement with target software developers
- ICON on future developments in NONMEM
- Pharsight considering using MDL to enhance PML
- Metrum on implementation with BUGS.

Valuable discussion and input from DDMoRe participants
- Subject matter experts contributing to MCL language features & TEL task definitions.
Modelling Object Group

Model
- INPUT_VARIABLES
- STRUCTURAL_PARAMETERS
- VARIABILITY_PARAMETERS
- GROUP_VARIABLES
- random
- RANDOM_VARIABLE_DEFINITION
- MIXTURE
  GROUP_VARIABLES
- OUTPUT_VARIABLES

Data
- HEADER
- INLINE
- RSCRIPT
- DESIGN

Parameter
- HEADER
- INLINE
- RSCRIPT
- DESIGN

Task Properties
- HEADER
- INLINE
- RSCRIPT
- DESIGN

Modelling Object Group (MOG)

- MOG objects
  - Model, parameter, data, task properties
  - Required inputs to TEL task object
  - The “model” is the MOG
  - User may combine objects from Repository MOG with user objects
    - E.g. Repository model+parameter with user data +task properties

- MOG Types
  - Defined & curated, static, { public }
  - User defined, read/write, { private, group, public }

Some Ways to Use a MOG

- Full Model (D,P,M,T)
  - Run the model to verify previous results
- Model (M)
  - Library call for model predictions (mixed effect)
  - User supplied D, P, T
- Simulate(M,P)
  - User supplied D and T
- Estimate(D,P,M)
  - User supplied estimation T using library D
- Data Transform(D)
  - User supplied T to transform library D

D=data, P=parameter, M=model, T=task_properties
A one compartment model with first-order input and first-order elimination. Dose is administered to compartment zero. Central compartment is 1 even if input is changed to zero-order or bolus.

```
LIBRARY {
F=PK(input=first-order, distribution=1, elimination=first-order, parameterization=vol=k,
param=list(
  cl=CL, v=V, DCMT=0, # input (depot) compartment is 0
tlag=ALAG0, ka=KA
) )
CONC=F.A1/V
}
```

A one compartment model with bolus input and parallel first and mixed-order elimination. The first-order elimination pathway is the formation route for a metabolite. The metabolite disposition is described by two compartments with mixed-order elimination. The metabolite has a delayed effect described by an effect compartment linked to the metabolite compartment.

```
LIBRARY {
F=PK(input=bolus, distribution=1, elimination=parallel-first-mixed-order, metabolite-formation=first-order, metabolite-distribution=2, metabolite-elimination=mixed-order, metabolite-link=effect, parameterization=vol=k, 
  param=list(
    v1=10, clfo=1, vmax=3, km=1, # parent
    FCMT=1, # metabolite is formed from 1st compartment of parent
    clsm=clfo, # Assume all first-order parent elimination leads to metabolite formation
    v1m=10, vmaxm=2, km=0.1, v2m=100, cl2m=4, # metabolite
    LCMTm=1m, teqm=1, # effect is determined by linking to metabolite compartment 1
  ) )
}
```

cem=F.effectm # effect compartment concentration of metabolite

effect=E=emax*cem/(c50+cem) # effect of metabolite
Model Coding Language

Rosetta Stone

Nick Holford and Mike K Smith on behalf of the DDMoRe consortium

Acknowledgement: The research leading to these results has received support from the Innovative Medicines Initiative Joint Undertaking under grant agreement n° 115156, resources of which are composed of financial contribution from the European Union’s Seventh Framework Programme (FP7/2007-2013) and EFPIA companies’ kind contribution. The DDMoRe project is also financially supported by contributions from Academic and SME partners.