Rosetta Stones in Population Pharmacokinetics and Pharmacodynamics

Rich Haney, Cellular Statistics, LLC

Introduction

1. In this work we look at the practical and also theoretical aspects of Rosetta Stones and their use in Population PK/PD. By a "Rosetta Stone" problem in this case we mean a problem of translating among data in different forms and models in different languages.

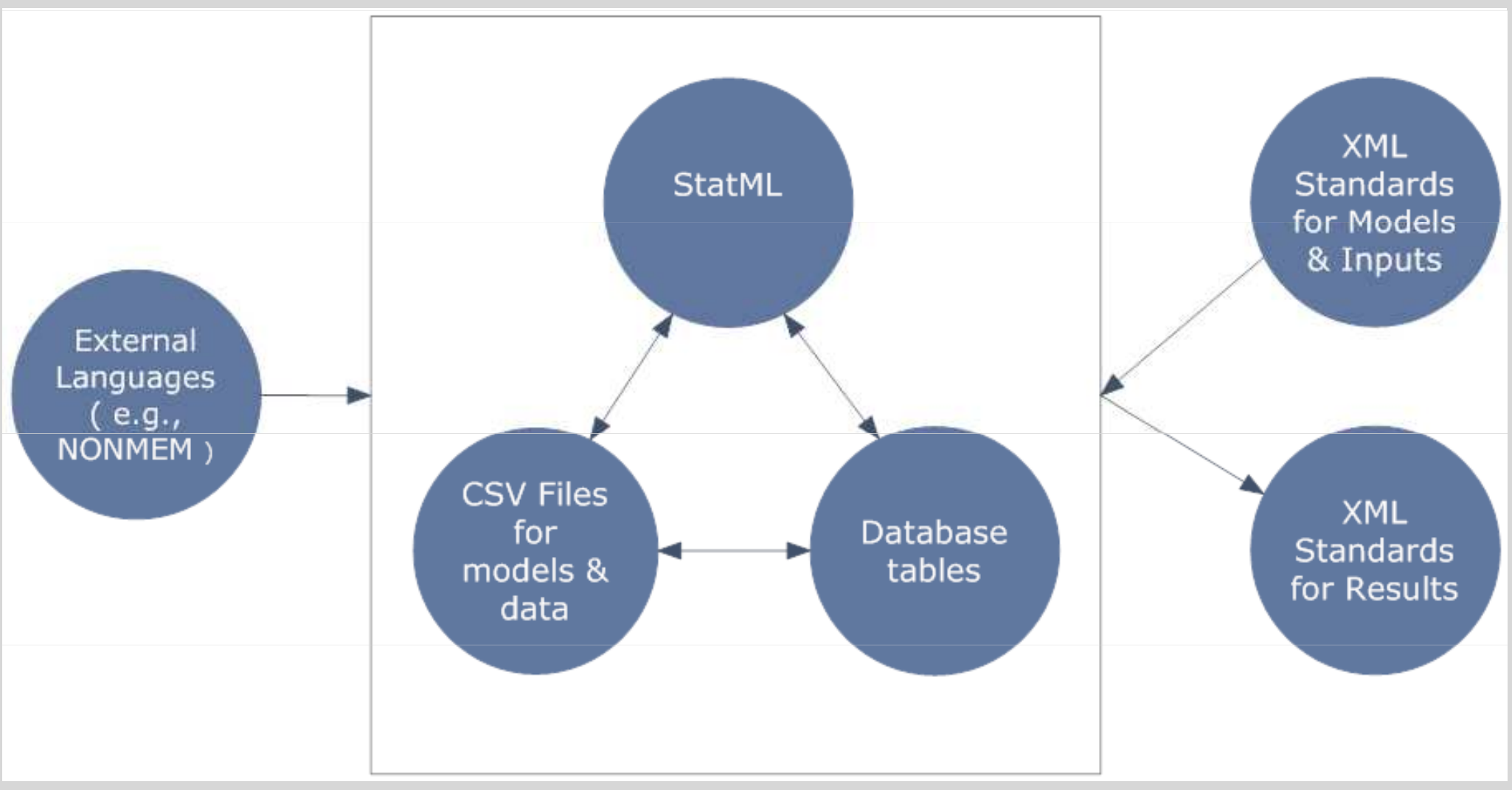
2. We make use of a "key" among languages that is not a language, but is algebraic in nature. In this context, we convert models to systems of equations that primarily involve the use of polynomials.

3. We hope to add to the public domain models originally in other languages (e.g., NONMEM) translated to Monolix.

4. We hope to be of assistance in deploying new standards or new platforms by companies such as the Pharsight Corporation

StatML - Files of equations for data and models

1. We use "StatML, a very simple form of XML in which we store vectors and equations in the same file. We use StatML for problems that involve Pop PK/PD and proteomics applications together. The advantage is simplicity, flexibility, and modifiability. For Pop PK/PD, as the diagram below shows, we convert from NONMEM to StatML and then other representations as follows:


StatML
External Languages (e.g., NONMEM)
CSV Files for models & data
Database tables
XML Standards for Models & Inputs
XML Standards for Results

2. The files of equations include models, data, and "meta-level" descriptions of modeling processes. In terms of syntax, individual equations do not use AND, OR, IF, THEN, ELSE, FOR LOOPs, or DO. In order to do proteomics as well as Pop PK/PD, the types of equations and other vocabulary are given in a separate file.

3. Most of the equations are in polynomial or "polynomial-type" forms called "polynomial functors". As such, we emphasize function composition, +, *, ^ (<=), D (for differentiation), and ` (for transpose or "adjoint"), and other algebraic operators.

4. Much of our handling of polynomials is done using Maple. For example, using Maple and polynomials, we handle " Birch criteria" of categorical data using Maple. We also create MathML, Latex and other representations of our models using Maple.

StatML - Sample Lines of Models & Data

In order to represent sets of models and data together (as so-called "algebraic varieties", and, more generally, "equational varieties"), we use about 12 types of equations overall. The following includes some different equations and examples of other equations as well:

```
AMT(1):    AMT    <INPUTS><- [ 60000 60000 60000 60000 60000 ..]
DV(1):     DV     <OBS>    <- [ 38.5 100.7 ..246.8 1469.8 05.5 2.1] …
DV(1):     DV     <PRED>  <- [ 39.1 101.9 ..246.8 1489.8 05.5 2.3] …
ln ( DV<OBS> )             <- LNDV(<OBS>)
InitialValues<A>           <- [A1(0) = AMT , A2(0) = 0]
DifferentialEqs<A>         <- [ DA(1) = KA*A(1) ,  DA(2)=KA*A(1)…]

DifferentialEqs<A>         <- [KA*A(1) +6000*(∂(72)+ ∂(106)+∂(120)…
 Integrals<A>              <- DA1(t) = AMT*exp(-KA*t)  x ... %implied
 IPred                     <- .001 + H(F)*(log(F) – .001 ) ( as in BUGS)
```

In this approach, we translate NONMEM, BUGS and other models to StatML, viewed as equations of different types, and for which equation types and other vocabulary are given separately. We then convert back from StatML to practical languages such as NONMEM. We also convert StatML to formal languages, primarily typed lambda calculus.

Rosetta Stones – some practical aspects

1. There are many examples of these for financial and costing systems. For example, one program (also named "Rosetta Stone") in the past has been very good at converting back and forth among different representations of software cost models in different formats. (Those have been cost models obtained using nonlinear mixed effects analyses.)

2. Many Computer Aided Software Engineering (CASE) tools since the 1970s do this for data and/or languages. Typical features include automated construction of a user interface from an algebraic forms such as database schema and data dictionary. However, these are typically not formally defined.

Rosetta Stones - some mathematical aspects

1. As implied in the last note, taking a mathematical approach to Rosetta Stones is not crucial, but we think it has some benefits.

2. One perspective is that "key" or *lingua franca* (common language), for our Rosetta Stone is actually not really a language at all. Instead, it is their common algebraic structure or algebraic specification.

3. In accord with theorems from algebraic logic, it is helpful for our StatML to make use of various additional versions of our files containing equations. The different versions "relate" to one other. Therefore, instead of having datasets, we have "dataset algebras". Instead of having models, we have "algebras of models".

4. In accord with #3, we create various other versions of StatML files of equations. In fact, we find that each version has its own practical value. We see theory and practice as very much aligned here.

Our own Rosetta Stone – some details

From the algebraic form, we then generate models in different languages. Here, we give here some details on our translator to BUGS from NONMEM. (Continuing our use of algebraic notation, we name our translator BUGS<=NONMEM, or BUGS^{NONMEM}.)

1. A NONMEM ADVAN1 or ADVAN2 type of model with single dosing yields

```
 for (i in 1:n.ind) {
      for (j in 1:nTimes ) {
          DV[i,j] ~ dnorm(ipredwb[i,j], tauC)
          ipredwb[i,j] <- DOSE[i]*exp(-CL[i]/V[i]*time[j])/V[i]
          ppredwb[i,j] <- DOSE[i]*exp(-exp(mu[1])/exp(mu[2])*time[j])/exp(mu[2])
}
…      beta[i, 1:2] ~ dmnorm(mu[ ],tau[,])
   }    …
mu  [1:2] ~ dmnorm(mean[ ], prec[ , ]);    tau [1:2,1:2 ] ~ dwish(R[ , ], 2); itau[1:2,1:2]<-inverse(tau[ , ])
```

2. In rare instances, when converting a NONMEM model to BUGS, an integral form is found that makes use of the LambertW function

```
     for (j in 1:nTimes ) {
.       DV[i,j] ~ dnorm(ipredwb[i,j], tauC)
         subExpr2[i,j]  <-  time[j]*VM[i]-KM[i]*V[i]*log(DOSE[i])-DOSE[i]
         subExpr1[i,j]  <-  exp(subExpr2[i,j]/(KM[j]*V[j]))/(KM[j]*V[j])
        ipredwb[i,j]    <-exp(-
    M[i]*V[i]*LambertW(subExpr1[i,j])+subExpr2[i,j])/(KM[i]*V[i]))/V[i]
 .
```

To run this type of model, , we implemented a PASCAL LambertW function using WinBUGS and Blackbox 1.5. (Blackbox 1.6 on Vista does not work for us.)

3. Multiple Dosing models work, but are not pretty:

```
        ipredwb[i,j] <- DOSE[i]*(step(tMinus72[j])exp(...-
        CL[i]*(tMinus72[j])/V[i])+step(tMinus144[j])exp(-[ .. this is 3 lines long]….
```

4. ODE versions as follows really need to be replaced by much faster PASCAL versions that we would automatically generate:

```
    ipredwbDES [1:nTimes, 1:n.Compartments] <-  ode(init[i,1:n.Compartments],
    time[1:nTimes], D(Subject[i,1:n.Compartments], t), 0, tol)
    D (Subject[i,1],t)       <-  -K[i] * Subject[i,1]…
```

One limitation: the author is "good" but not "expert" as a Bayesian statistician.

A Note on BUGS<=NONMEM

Given space and time, we would want to try to persuade you that:

1. If |NONMEM| is the number of parts in a NONMEM model, and |BUGS| is the number of parts in a BUGS model, then $|\text{BUGS}|^{|\text{NONMEM}|}$ is one index of how hard the problem is.

2. It is in this context reasonable to use the notation above in which we named this particular tool BUGS<=NONMEM (or, BUGS$^{\text{NONMEM}}$)

Rosetta Stone translations - having "Round trips"

One method to ensure that the many kinds of translations are working properly involves making additional round trips. If T means to go to BUGS, and T' (for the adjoint of T), then T' * T means we are going from NONMEM to BUS and then back again.

So, if we start from NONMEM and go to BUGS, we ultimately want:

T = T * T' * T

It just means that the result of translation should stay the same if we do one more "round-trip" to NONMEM. It's just easier to check.

Round trips, monads, and higher-level operations

1. In the model theory of differential equations, we would be apt just to call such a round-trip a "back-and-forth" (or, more precisely for this case, and "up-and-down" operator).

2. In universal algebra, we would consider round trips as "monads"

3. Category theory greatly enriches the notion of monads through analysis of these structures as pairs of adjunctions.

Since the notion arises within algebra in three different ways, we think it gives extra weight to the idea of doing such "round trips" in practice.

By building upon notions of monads and other mathematical aspects of Rosetta Stones, we can then add higher-level operations as well. This is as part of our overall work on "Open Statistical Services".

Partial Summary of Results

1. We have translated a large number of models back and forth among languages including NONMEM, BUGS, Maple, and Monolix. We have probably done about 600 individual translations from some given source to some given target. However, to be really complete, we would need to do 5-10 times that number.

2. We were able to do about 80 round trips properly.

3. Our use of BUGS originally made use of the final estimates from NONMEM runs. In fact, this is how we would tend to use BUGS in general. On the other hand, we need to test all our BUGS models using the same initial estimates originally given to NONMEM.

4. Maple was extremely effective as a tool. However, use of Maple of coding up translation to polynomial form of categorical data models was not helpful as we had hoped. (We have not had true full categorical data models to use, though.)

5. We easily generated Monolix "project" files in MATLAB, stored separately from Monolix structural models. With respect to generation of Monolix structural models, we had some difficulty with generating multiple dosing models & new types of ODEs.

Conclusion

The "Rosetta Stone" problem in Pop PK/PD is one for which an algebraic perspective is not crucial. However, use of an algebraic perspective does provide one solution to the problem that is quite clean. It yields some additional insights that we think also have some practical benefits. Finally, it fits in overall with an approach in which we consider model building processes as algebraic in nature, too.

References

[1] Rich Haney, Informal presentation at SAS, April 2004
[2] "Interoperability of MATLAB, SAS, and R: Formal Methods and Practical Aspects", SUN Computational Biology Conference, September 22, 2005
[3] Peter McCullagh, "What is a statistical model?", The Annals of Statistics, 2002, Vol. 30, No. 5, 1225–1310
[4] Seth Sullivant, "Statistical Models are Algebraic Varieties", 2006, http://www.math.harvard.edu/~seths/lecture1.pdf
[5] Rich Haney, "Category theory for use in formal systems design for biological applications: useful approach or Siren Song?", 1985.

Acknowledgements

The concept of Open Statistical Services and StatML was partly in response to a meeting with the NONMEM consortium in September 2004 in which the author was involved. Our earlier work on StatML was with Patrick McConnell. For that we are also thankful for funding from caBIG.
Thor Mirchandani & Brad Sparks of Presens Technologies did our database & model comparison tool. This allows us to compare hundreds of models to one another.