An algorithm for proper lumping in ODE systems

Aris Dokoumetzidis and Leon Aarons
Centre for Applied Pharmacokinetic Research, University of Manchester

Objectives

We present an algorithm for automatic order reduction of large mathematical models, with ultimate intention of using it to simplify systems biology models with drug interest, such that they can be used as pharmacodynamic models.

Methods

A nonlinear system of first order ordinary differential equations (ODE) is written as:

$$\frac{dy}{dt} = f(y) \quad (1)$$

where y is a vector of n in number states.
We want to reduce the system producing a new set of differential equations with a new set of states $\tilde{y}$, $\tilde{n}$ in number, where $\tilde{n}<n$, such that some of the new states $\tilde{y}$ are lumps of some of the original states y, while the rest are left unlumped. The new lumpd model can be writen as:

$$\frac{d\tilde{y}}{dt} = \tilde{f}(\tilde{y}) \quad (2)$$

Lumping can be writen by using an $\tilde{n}$ by n lumping matrix M, comprising of 1s and 0s that determines which states are lumped as follows:

$$\tilde{y} = My \quad (3)$$

This type of lumping is called “proper” because each state of the original model may be included in only one lump. This is opposed to more general lumping schemes where the lumps are linear combinations, or even nonlinear functions, of several of the original states, which means that they loose their physical meaning. Given this matrix M it is easy to show that a good choice for the lumped model is given by the following relation (1).

$$\tilde{f}(\tilde{y}) = Mf(M^{+}\tilde{y}) \quad (4)$$

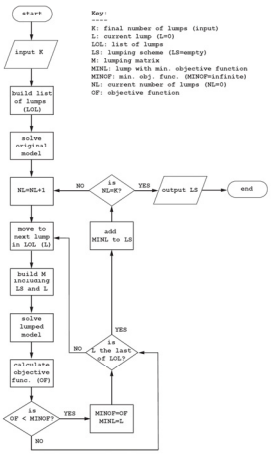
where M^{+} is the pseudo-inverse of M, such that $MM^{+}=I$.
The lumping algorithm employed here, given a model $f(y)$ and a desired order $\tilde{n}$, considers all possible combinations of lumped models, runs them and compares each solution to the solution of the original model based on an objective function

$$d = \sum_{i=1}^{\tilde{n}} \int_0^T \left(\frac{y_i - \tilde{y}_i}{\int_0^T y_i dt} \right)^2 dt \quad (5)$$

The objective function d is basically the sum of squares of the distances of all the states of the reduced and the original model, normalised by the average of the solution of the original model.
The algorithm then returns the lumping scheme which has the smaller value of the objective function.
A problem of such a search is combinatorial explosion. Given a model of dimension n, the number of possible lumped models of dimension one less, i.e. n-1 is $n(n-1)/2$. So if we want k lumps in total, which means a final model of dimension n-k, the total number of combinations is given by the formula:

$$\frac{(n(n-1)/2)!}{k!(n(n-1)/2-k)!} \quad (6)$$

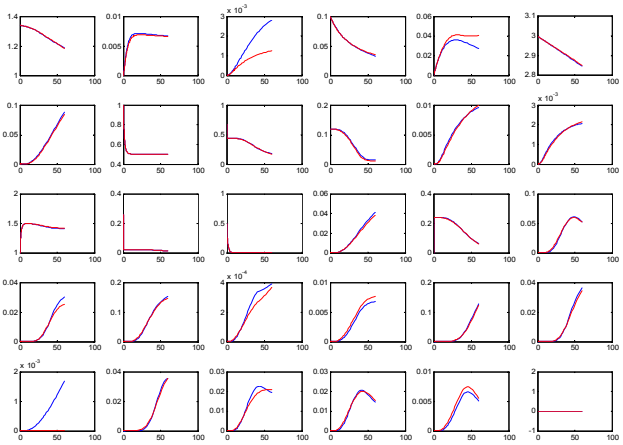
So for example for n=30 and k=20 the total number of combinations is about 10^{35}, which basically makes the direct search of all combinations impossible.
However not all combinations need to be considered. If we find the best lumping scheme for only one lump, which is not very expensive to do ($n(n-1)/2$ = 435 combinations for n=30) then the best lumping scheme for two lumps will include the first lump already found, plus one more (another 434 combinations) and we can repeat the procedure estimating one lump at a time. This means that we can estimate all k lumps starting from the fist one and move on one by one, examining a total number of less than $k \cdot n(n-1)/2$ combinations (namely 8490 combinations for n=30 and k=20) which is perfectly feasible. This type of algorithm is often referred to as a “forward selection” algorithm.



The implementation of the algorithm outlined above was done in MATLAB and it is shown in the flow chart.

Results

In order to see the algorithm in action we downloaded from the BioModels database (http://www.ebi.ac.uk/biomodels/) a fairly large model in SBML (Systems Biology Mark-up Language) which describes the ERK signalling network (2). This model includes 99 species and 150 reactions. SBMLToolbox (http://sbml.org/software/sbmltoolbox/) was used to translate the SBML file to a MATLAB function.
In the following figure a simulation of the reduced model (red lines) and the original model (blue lines) of an indicative lumping scheme, produced by the algorithm, is shown, corresponding to a reduction from 99 species down to 30.



The simulations in the figure show that although the model is significantly reduced, the kinetic behaviour of most species is preserved (exceptions are species 3, 5 and 25). The lumping of the this model included around 340000 function calls and took nearly 9 hours to run on a Intel Xeon 2.66 MHz processor, using MATLAB 7.4.

Discussion and Conclusions

The lamping algorithm presented here has several favourable features:

- **It is completely automatic and easy to use.** One may start off with a SBML file of a model one knows nothing about and easily use the method reducing the model significantly.
- **Handles nonlinear models.** Unlike other methods that are based on linear algebra techniques and are very difficultly applied for nonlinear models this method makes no discrimination between linear and nonlinear models. This is an important feature as most biochemical systems have nonlinearities.
- **Allows constraints.** One can easily implement constraints such as force specific states to be left unlumped, or exclude specific lumping combinations without interfering with the main code.
- **Allows flexibility in the objective function.** The objective function of Eq. 5 is only indicative. One may consider functions that compare different features of the responses of the models, which may be more appropriate for other cases, e.g. when oscillations are present. Also Eq. 5 assumes that there is an equal interest in all states of the model. This is unlikely, so one could introduce higher weights on the most interesting states.
- **Parameter uncertainty may be included.** Similar to the pervious point, one may easily implement a Bayesian objective function that averages over prior parameter distributions. Such an objective function could look like:

$$d_{Bayes} = \int_{prior} P(\theta) \log(d) d\theta \quad (7)$$

- **Can be easily parallelised.** The algorithm may be converted to run over a cluster of distributed processors in a straightforward manner. This may be particularly important when parameter uncertainty is included as described in the previous point.

Overall the presented algorithm is an easy to use and quite efficient method for lumping in ODE systems, which is expected to be a useful tool towards our general aim of developing methodology to link systems Biology Models and PK/PD.

1. Sasagawa S, Ozaki Y, Fujita K, et al. Prediction and validation of the distinct dynamics of transient and sustained ERK activation. Nature Cell Biology 7 (4): 365-373 (2005).
2. Li G, Rabitz H. A general-analysis of exact lumping in chemical-kinetics. Chemical Engineering Science 44 (6): 1413-1430 (1989).

Correspondence: School of Pharmacy and Pharmaceutical Sciences, The University of Manchester, Stopford Building, Manchester, M13 9PT, U.K. **Email**: dokoumetzidis@manchester.ac.uk.