Parallel Bayesian Methodology for Population Analysis

Ihab G. Girgis[1,4], Tom Schaible[2], Partha Nandy[3], Filip De Ridder[1], Jeffrey Mathers[2], and Surya Mohanty[1]

Johnson & Johnson Pharmaceutical Research & Development

[1]Advanced Modeling & Simulation, Biometrics and Clinical Informatics
[2]Technology Office
[3]Advanced Modeling and Simulation, Clinical Pharmacology
[4]igirgis@prdus.jnj.com

Introduction

Present processors are fast and powerful to run most of the statistical computations within reasonable CPU time. However, when using sophisticated models and large datasets some simulations may require days to compute. Parallel computing could be a very efficient approach to test and compute different scenarios and explore diverse suite of physiological and statistical models in robust and time efficient way. Nevertheless implementing parallel computation is not trivial.

Objectives

The aim of this work is to present a novel computational tool to expedite population pharmacokinetics analysis (PK) and perform modeling and simulation using fully parallel Bayesian approach through Markov chain Monte Carlo (MCMC) technique. This tool can be applied to perform population pharmacodynamics (PD) and population PK/PD analysis as well.

The proposed computational capability could be efficiently used for executing parallel analysis, such as, model building and selection, covariates searching, convergence testing, and initial-value independence check, as shown in Figures 1 and 2.

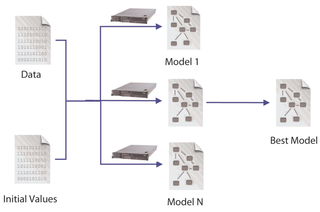


Figure 1. GridBUGS Implementation for Model and Covariate Selection

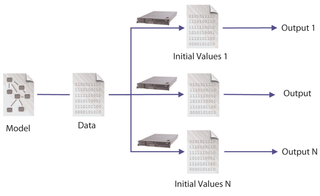


Figure 2. GridBUGS Implementation to Check Convergence and Initial-Value Independence

Since Bayesian computation could be subdivided into components that would be done independently, this tool provides a unique opportunity for executing a single model in parallel on large number of independent processors and then collect and merge each processor's statistical output to obtain the final model estimates, as shown in Figure 3.

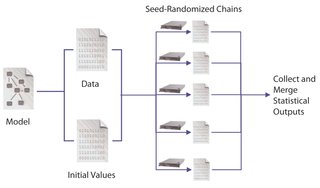


Figure 3. GridBUGS Implementation for Multiprocessor Execution of a single Model

This tool uses WinBUGS to fit a wide range of complex differential and closed-form models with a high degree of flexibility capable of handling multiple levels of uncertainty (variability), missing data, and time discontinuities.

Methods

A web application, GridBUGS, was built and tested to execute WinBUGS 1.4[1], with the WBDiff add-on[2], on Johnson & Johnson's grid[3], which enables using hundreds of high-end dedicated machines, with processing power that exceeds 1200 Gflops.

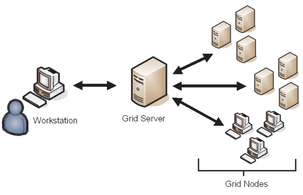


Figure 4. Johnson & Johnson's Grid Architecture

Methods (cont'd)

The web interface was built using a Java based client framework that provides a rich graphical user interface. The interface provides many useful utilities for submitting Bayesian models with multiple input files such as script generation, input editing, automatic input checking, execution concurrency, and chain randomization. Additionally, it provides tools for monitoring and retrieving submitted models, such as customizable downloads, early result retrieval, and auto-refreshing tables.

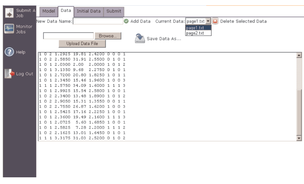


Figure 5. GridBUGS Screen Snapshot for Uploading Multiple Input Data Files

Figure 6. GridBUGS Screen Snapshot for Managing Randomized Chains

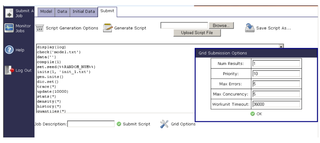


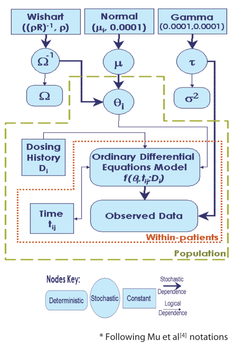
Figure 7. GridBUGS Screen Snapshot for Submission Options

Figure 8. GridBUGS Screen Snapshot for Retrieving Results

Study Case and Results

We illustrate the implementation of this tool by analyzing a PK dataset and comparing the results with the estimates obtained through the NONMEM software.

A PK dataset was simulated using three-compartment closed form model with infusion input for different doses. A log-normal error model was assumed and both inter-individual and intra-individual variabilities were considered. Differential equations describing the three-compartment model were then used to fit the above data using both GridBUGS and NONMEM (FO method). A very good agreement was observed between the two results.



* Following Mu et al[4] notations

Figure 9. Statistical Model*

Study Case and Results (cont'd)

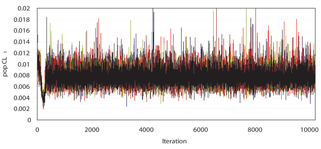


Figure 10. Five Randomized Chains History for Clearance Parameter (all chains were merged and used to construct the final solution)

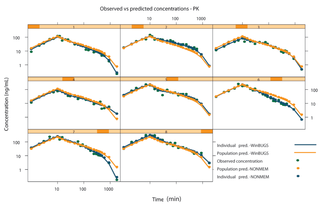


Figure 10. Observed vs. Predicted Concentrations

Table 1. Population Parameters Estimate

	True*	WinBUGS – mean values (95% credible interval symmetrical)	NONMEM_FO method (95% credible interval based on CL%SE)
V_1 (L)	**0.588**	**0.585** (0.427,0.788)	**0.591** (0.484,0.698)
V_2 (L)	**1.871**	**1.763** (1.012,2.678)	**1.700** (1.185,2.215)
V_3 (L)	**0.575**	**0.689** (0.351f,1.239)	**0.731** (0.425,1.037)
CL_1 (L/h)	**0.00769**	**0.00773** (0.005551,0.01049)	**0.00691** (-0.00838,0.02220)
CL_2 (L/h)	**0.01250**	**0.01065** (0.005653,0.01771)	**0.01040** (0.00723,0.0136)
CL_3 (L/h)	**0.05261**	**0.04663** (0.02257,0.08413)	**0.03850** (0.0246,0.0524)
η_{11} for V_1	**0.1**	**0.0687**	**0.0381**
η_{22} for V_2	**0.1**	**0.0791**	**0.1690**
η_{33} for V_3	**0.1**	**0.1693**	**0.1580**
η_{44} for CL_1	**0.1**	**0.0777**	**0.0750**
η_{55} for CL_2	**0.1**	**0.0643**	**0.0185**
η_{66} for CL_3	**0.1**	**0.0977**	**0.444**
σ	**0.224**	**0.221** (0.1979,0.2468)	**0.214** (0.00139)

* *Values used to generate the dataset*

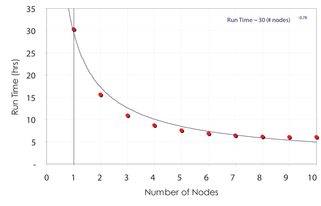


Figure 11. Run Time versus Number of Nodes for Present Study Case (2000 burn-in iterations and solutions of 200,000 iterations on 1.7GHz Pentium machined)

Conclusions

GridBUGS is a useful tool for model development that provides researchers with vast flexibility to expedite, explore and subsequently incorporate diverse statistical objectives in model building. This web-based application could be implemented both in parallel and serial methodologies to execute both differential and closed-form models. Although, expediting the computation is dependent on many factors, such as, nature of the model, size of the dataset, number of burn-in iterations, and number of the required iterations for convergence, for the present study case, an exponential reduction in run time was achieved as number of nodes increases. As shown in Figure 11, the optimum number of nodes (processors) for our study case was between three to six processors. Using three processes has resulted in 64% reduction in run time while using six processors resulted in 78% reduction. The asymptote value of run time reduction is around 80%.

References

[1] http://www.mrc-bsu.cam.ac.uk/bugs/

[2] http://www.winbugs-development.org.uk/wbdiff.html

[3] http://www.ud.com/products/gridmp.php

[4] Mu S, Ludden TM. Estimation of population pharmacokinetic parameters in the presence of non-compliance. J Pharmacokinetics Pharmacodynamics 2003; 30: 53-81