PhUSE 2007

hPaper ST07

Bayesian Modeling with S-PLUS® and the S+flexBayes Library

Michael O'Connell, Insightful Corporation, RTP, North Carolina, US
Dawn Woodard, Insightful Corporation, RTP, North Carolina, US
Joel Hoffman, Insightful Corporation, Basel, Switzerland
Andrew Jack, Insightful Corporation, Basingstoke, UK

ABSTRACT

Bayesian methods are used increasingly across all phases of the pharmaceutical research and development cycle. Examples include early-phase adaptive clinical trial design, population-pk modeling and analysis of safety data in all phases. These applications are driving the need for robust, flexible and novel Bayesian analytics, graphics and reporting software.

This article and presentation reviews Insightful Corporation's approach to providing Bayes analysis software tools and solutions to support the needs of pharmaceutical development. In particular, the Insightful S-PLUS® flexBayes library (S+flexBayes), in conjunction with the collaborative S-PLUS framework, provide statisticians and statistical programmers with a flexible and productive solution for Bayes modeling. Results from these analyses are readily deployed as interactive clinical review graphics or publication reports as part of the Insightful Clinical Review and Reporting Solutions. We illustrate the S-PLUS Bayes modeling and collaborative Clinical Review Solutions with an application of the Berry and Berry (2004) model for analysis of adverse events.

INTRODUCTION

Bayesian methods are used increasingly across phases of the pharmaceutical research and development cycle. Examples include early-phase adaptive clinical trial design, population-pk modeling and analysis of safety data at all phases. These applications are driving the need for robust, flexible and novel Bayesian analytics, graphics and reporting software.

This article and presentation reviews Insightful Corporation's approach to providing Bayes analysis software tools and solutions to support the needs of pharmaceutical development. In particular, the Insightful S-PLUS flexBayes library (S+flexBayes), in conjunction with the collaborative S-PLUS framework, provide statisticians and statistical programmers with a flexible and productive solution for Bayes modeling. Results from these analyses are readily deployed as interactive clinical review graphics or publication reports as part of the Insightful Clinical Review and Reporting Solutions. We illustrate the S-PLUS Bayes modeling and collaborative Clinical Review Solutions with an application of the Berry and Berry (2004) model for analysis of adverse events.

This paper is organized as follows: we first give a high-level introduction to Bayesian modeling in the context of clinical drug development. We then introduce S+flexBayes and the S-PLUS® software environment. The benefits of

Bayesian modeling using S+flexBayes are then illustrated with an application of the Berry and Berry (2004) model for analysis of adverse events. Implementation of the Berry and Berry model is shown in S-PLUS code, using S+flexBayes and links to the BUGS environment. The workflow for the example is described, including data pre-processing, model definition, sampling and post-processing; with relevant graphical summaries and deployment using the Insightful Clinical Review Solution.

BAYESIAN STATISTICS

In Bayesian statistics, information in the data and *a priori* information are combined into the *posterior distribution* of the parameters. This posterior distribution is used to infer the values of the parameters, along with the associated uncertainty. Multiple tests and predictions can be performed simultaneously and flexibly. Quantities of interest that are functions of the parameters are straightforward to estimate, again including the uncertainty. Posterior inferences can be updated as more data are obtained, so that study design is more flexible than for frequentist methods.

Bayesian inference is possible in a number of contexts where frequentist methods are deficient. For instance, Bayesian inference can be performed with small data sets. More broadly, Bayesian statistics is useful when the data set may be large but where there are few data points associated with a particular treatment. In such situations standard frequentist estimators can be inappropriate since the likelihood may not be well-approximated by a normal distribution.

Bayesian statistics also allows for the incorporation of prior information, and for simultaneous inference using data from multiple studies. Inference is also possible for complex hierarchical models, for example in models for pharmacokinetics/dynamics describing drug absorption, distribution, metabolism and excretion in/from the human body.

Computation for Bayesian models is most often done via Markov Chain Monte Carlo (MCMC) techniques, which obtain dependent samples from the posterior distribution of the parameters. In MCMC, a set of initial parameter values are chosen. These parameter values are then iteratively updated via a specially constructed Markovian transition. In the limit of the number of iterations, the parameter values are distributed according to the posterior distribution. In practice, after approximate convergence of the chain, the time series of sets of parameter values can be stored, and then used for inference via empirical averaging ("Monte Carlo"). The accuracy of this empirical averaging depends on the effective sample size of the stored parameter values; i.e. the number of iterations of the chain after convergence, adjusted for the autocorrelation of the chain.

One way of specifying the Markovian transition is via Metropolis-Hastings, which proposes a change in the parameters, often according to a random walk, and then accepts or rejects that move with a probability that is dependent on the current and proposed state.

In order to perform valid inference, the Markov chain must have approximately converged to the posterior distribution before the samples are stored and used for inference. In addition, enough samples must be stored after convergence to have a large effective sample size; if the autocorrelation of the chain is high then this may be a large number of samples. Detection of lack of convergence or high autocorrelation of the chain is done via convergence diagnostics, which include autocorrelation and trace plots, as well as Geweke, Gelman-Rubin, and Heidelberger-Welch diagnostics (Geweke 1992, Gelman and Rubin 1992, Heidelberger and Welch 1983).

Validation of software for MCMC can also be done using a distinct set of techniques, namely those developed by Geweke (2004) and Cook, Gelman and Rubin (2006). These techniques compare the posterior samples drawn by

the software to samples from the prior and the data model, thereby validating the joint distribution of the data and parameters as estimated by the software.

SOFTWARE

S+flexBayes is a software package allowing flexible Bayesian modeling within S-PLUS, an object-oriented environment for statistical computing and graphics. S-PLUS runs on multiple Windows and UNIX platforms, connects natively with a wide variety of data sources, and produces a comprehensive set of interactive and publication statistical reports.

S-PLUS differs from other statistical software in being largely composed of function objects, which can be openly examined and modified by the user. S-PLUS has a comprehensive set of statistical techniques, and produces high-quality and customizable statistical graphics: including Trellis™ graphics for visualizing multivariate data, and Graphlets™ for interactive metadata brushing and drill-down.

There are many modules, libraries, and packages that extend the functionality of core S-PLUS. In 2007, with the release of S-PLUS 8, a freely-available repository of S-PLUS packages was created (csan.insightful.com) to enable sharing of application code among S-PLUS users. S+Server™ enables S-PLUS analytics to be made available through browsers or custom application interfaces, including SAS®. Insightful offers a number of solutions for pharmaceutical companies based on S-PLUS and S+Server, for clinical statistics, graphics, reporting and review of clinical data.

S+flexBayes

Once the S+flexBayes package is loaded in the S-PLUS environment, the user specifies the model, prior distributions, and data set. Dependent samples from the posterior distribution are obtained by the package via a Markov chain algorithm. These samples are then analyzed in order to test the convergence of the chain and to perform posterior inferences via Monte Carlo. The S+flexBayes package also includes a suite of validation functions based on the methods of Geweke (2004) and Cook, Gelman, and Rubin (2006). This enables the package to be readily validated in the end-user environment.

Using S+flexBayes, inference can be performed for a very broad range of models. For instance, non-linear regression models such as Emax or four-parameter logistic models can be specified with prior distributions on the parameters.

Once the model and data are specified, posterior samples are obtained by S+flexBayes using one of several engines that are available for the package. These engines currently include the open-source software WinBUGS and OpenBUGS, and work is in progress to include the JAGS engine. Insightful also has its own internal engine in S+flexBayes that is tailored to the specific class of hierarchical generalized linear models.

For models of particular interest, specialized functions provide simplified interfaces and increased efficiency of sampling. For instance, non-linear models with mixed random and fixed effects and overdispersion can be fit very easily and efficiently using S+flexBayes.

Once the model has been fit, a broad range of statistical inference can be performed by the user, utilizing the posterior samples obtained by S+flexBayes. Graphical displays may be created to summarize results, using additional S-PLUS packages such as S+Safety, a library of analytics and graphics supporting analysis of drug safety

data. Results of these analyses can be readily deployed through the Insightful Clinical Review Solution for interactive medical monitoring and safety review.

CASE STUDY: HIERARCHICAL MODELING OF ADVERSE EVENT DATA

BACKGROUND

In drug safety data analysis, one key goal is early detection of any increased risk of an adverse event. A wide range of possible adverse events are monitored in early- and late- phase drug trials, and any one of these could occur with increased frequency in the treatment group. Detection of such treatment-emergent adverse events should be as early as possible but must also be carefully diagnosed, since the cost of spurious identification is expensive and lengthens the drug approval process. Bayesian methods are valuable in this area since inferences can be performed incrementally as data are collected, and since valid inferences can be performed even for very rarely-occurring adverse events. In addition, improved detection over frequentist tests is possible due to the ability to borrow strength across the set of adverse events. Bayesian models can also automatically adjust for multiple testing, as shown below.

THE BERRY AND BERRY (2004) MODEL FOR ANALYSIS OF ADVERSE EVENT COUNTS

Berry and Berry (2004) describe a three-level hierarchical mixture model for analysis of adverse event counts. In addition to the usual Bayesian hyper-parameter hierarchies, the model exploits hierarchies in adverse event coding structure namely the nesting of preferred terms (PT) within system organ classes (SOC).

For each system organ class (i) and each adverse event within that system organ class (i,j), the number of occurrences of adverse events in control and treatment are denoted X_{ij} and Y_{ij} respectively. These are assumed to be distributed binomially, with the risk parameters for the control and treatment group being denoted c_{ij} and t_{ij} respectively. The ratio t_{ij}/c_{ij} is called the "relative risk" of the adverse event in treatment versus control. The parameters c_{ij} and t_{ij} are transformed via a logistic link:

$$\gamma_{ij} = \text{logit}(c_{ij})$$
$$\theta_{ij} = \text{logit}(t_{ij}) - \gamma_{ij}$$

Here, the parameter θ_{ij} is a logodds ratio for treatment versus control. The control-group parameter is then assigned a normal prior, while the elevated-risk parameter is given positive probability of being equal to zero, and otherwise is distributed normally:

$$\gamma_{ij} \sim N(\mu_{\gamma i}, \sigma_{\gamma}^2)$$
$$\theta_{ij} \sim \pi_i I_{[0]} + (1-\pi_i) * N(\mu_{\theta i}, \sigma_{\theta}^2)$$

The parameter π_i is the probability that there is no treatment effect for a (new) adverse event in system organ class i. Since the parameters $\mu_{\gamma i}$, σ_{γ}^2, π_i, $\mu_{\theta i}$, and σ_{θ}^2 are the same for all AEs within a single SOC, this level of the hierarchy allows borrowing of information within the SOC. The parameters $\mu_{\gamma i}$, $\mu_{\theta i}$, and π_i are then modeled as random effects:

$$\mu_{\gamma i} \sim N(\mu_{\gamma 0}, \tau_{\gamma}^2)$$
$$\mu_{\theta i} \sim N(\mu_{\theta 0}, \tau_{\theta}^2)$$
$$\pi_i \sim \text{Beta}(\alpha_{\pi}, \beta_{\pi})$$

Here we have simplified the model slightly, since Berry and Berry (2004) allow σ_θ^2 to be SOC-specific. Since the parameters $\mu_{\gamma 0}$, τ_γ^2, $\mu_{\theta 0}$, τ_θ^2, α_π, and β_π are the same for all SOCs, this level of the hierarchy allows borrowing of information across the SOCs. This specification also gives non-trivial prior weight to the possibility that few or no AEs are affected by the treatment, which addresses the multiple testing issue. This is done in a refreshingly direct manner by focusing the model on the two sub-populations of interest i.e. adverse events that are affected by treatment and those that are not.

The prior distributions for the remaining parameters are then specified as in Berry and Berry (2004):

$\mu_{\gamma 0} \sim N(\mu_{\gamma 00}, \tau_{\gamma 00}^2)$
$\mu_{\theta 0} \sim N(\mu_{\theta 00}, \tau_{\theta 00}^2)$
$\tau_\gamma^2 \sim IG(\alpha_{\tau\gamma}, \beta_{\tau\gamma})$
$\tau_\theta^2 \sim IG(\alpha_{\tau\theta}, \beta_{\tau\theta})$
$\sigma_\gamma^2 \sim IG(\alpha_{\sigma\gamma}, \beta_{\sigma\gamma})$
$\sigma_\theta^2 \sim IG(\alpha_{\sigma\theta}, \beta_{\sigma\theta})$

The parameters α_π and β_π are given independent exponential prior distributions, with hyperparameters λ_α and λ_β respectively, truncated below by 1.

Note that $\mu_{\gamma 00}$, $\tau_{\gamma 00}^2$, $\mu_{\theta 00}$, $\tau_{\theta 00}^2$, $\alpha_{\tau\gamma}$, $\beta_{\tau\gamma}$, $\alpha_{\tau\theta}$, $\beta_{\tau\theta}$, $\alpha_{\sigma\gamma}$, $\beta_{\sigma\gamma}$, $\alpha_{\sigma\theta}$, $\beta_{\sigma\theta}$, λ_α and λ_β are constants. Choices of these constants relate to how informative the user would like the prior distribution to be; in Berry and Berry (2004), these values are chosen such that the prior distributions are fairly vague. The λ_α and λ_β values govern the prior proportion of adverse events affected by the treatment. In our implementation of the model we study the level of sensitivity of the posterior to changes in the prior distributions, by changing the values of these constants.

CODE ELEMENTS FOR MODEL FITTING

The code for running and analyzing the Berry and Berry model is available in the `drugAdverseEvents` help file in the S+flexBayes library. Here, we show the important parts of this code.

First, the data are read in. The data set analyzed in this paper is available in the S+flexBayes package, and is loaded with the code

```
> data(AE.PTwSOC)
```

Preprocessing of the data is then done to create a data object called `aeCounts`. Next, the model is fit using the specialized S+flexBayes function `runBerryAndBerry`:

```
> aeSimNotDiffuse <-
  runBerryAndBerry(aeCounts, nIter = 4000, nThin = 10, diffusePrior = F, engine = "OpenBUGS")
```

Here, the user has chosen to obtain 4000 samples from the posterior distribution, after thinning by 10 to reduce autocorrelation of the samples. The user has also chosen to use the hyperparameter values from Berry and Berry (2004), rather than values that lead to more diffuse prior distributions. Note that this is set in the arguments of the runBerryAndBerry() call. This approach of exposing (prior) parameters in the S-PLUS function that calls the BUGS engine makes it very convenient to fit and assess models from S-PLUS, without having to manually specify the BUGS code for the model or the options for the inference engine.

This function call draws the posterior samples using the OpenBUGS engine. Another engine, e.g. WinBUGS, can be chosen by changing this argument setting. The resulting `aeSimNotDiffuse` object is an object of class `posterior`, which is defined and handled by a set of methods in the S+flexBayes package.

The code of the `runBerryAndBerry` function illustrates the workflow of running a model within the flexible framework of S+flexBayes, and is described in the Appendix.

CODE ELEMENTS FOR DIAGNOSTIC AND RESULTS PLOTTING

A wide array of convergence diagnostics is available in S+flexBayes for the `posterior` object. Since there are a large number of parameters in the Berry and Berry model, we perform the diagnostics on a representative subset of parameters:

```
> diagnosticParameters <- aeSimNotDiffuse[,c("c[3,1]", "a[3,1]", "psi.a[3,1]", "mu.c[3]",
          "s.rho.a", "s.rho.c", "s.tau.c", "tau.a[1]", "tau.a[3]", "theta.t[3,1]",
          "mean.c", "phi.a[1]", "phi.a[3]", "alpha.phi", "beta.phi", "a[5,3]", "c[5,3]", )]
> traceplot(diagnosticParameters)
> autocorr.plot(diagnosticParameters)
> effectiveSize(diagnosticParameters)
> geweke.diag(diagnosticParameters)
> geweke.plot(diagnosticParameters)
> gelman.diag(diagnosticParameters)
> gelman.plot(diagnosticParameters)
> heidel.diag(diagnosticParameters)
```

The Geweke-Brooks plot for six parameters is shown in Figure 1.

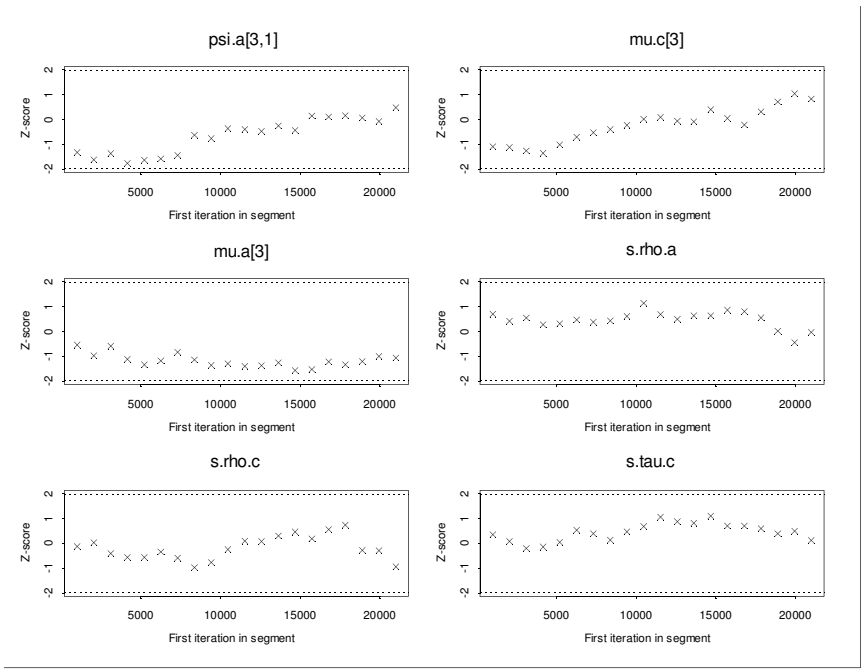


Figure 1. The Geweke-Brooks diagnostics plot for six chosen parameters. This graph is exported from the S-PLUS graphsheet() device as a WMF file. This vector graph format can be resized in the MS Word document without affecting the graphic quality.

For this example, Z-scores are within allowable ranges, and the diagnostics do not detect any problems with lack of convergence or high autocorrelation of the posterior samples. The samples are then used for inference; the simplest inference tool is the summary function:

```
> summary(aeSimNotDiffuse)
```

which yields

```
Iterations = 1001:40991
Thinning interval = 10
Number of chains = 1
Sample size per chain = 4000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                Mean       SD  Naive SE Time-series SE
     s.tau.c 0.98632 0.082994 0.0013122      0.0013599
     s.rho.a 0.49835 0.106507 0.0016840      0.0020295
     s.rho.c 0.46894 0.094951 0.0015013      0.0014702
theta.t[1,1] 0.02178 0.009302 0.0001471      0.0001435
theta.t[2,1] 0.01338 0.007416 0.0001173      0.0001045
…

2. Quantiles for each variable:

                 2.5%    25.0%   50.0%   75.0%   97.5%
     s.tau.c 0.835491 0.928126 0.98231 1.03843 1.15585
     s.rho.a 0.334737 0.424329 0.48448 0.55516 0.75750
     s.rho.c 0.320313 0.402427 0.45656 0.52354 0.67840
theta.t[1,1] 0.007174 0.014860 0.02066 0.02743 0.04323
theta.t[2,1] 0.003410 0.007905 0.01207 0.01734 0.03126
…
```

Additional graphical summaries are available for the Berry and Berry model using the S+Safety library, also available from Insightful. After loading that library, the user can call:

```
> pRiskPlot(aeSimNotDiffuse, treatmentRiskParam = "theta.t",
          controlRiskParam = "theta.c", … )
```

This yields a pRisk plot (O'Connell, 2006), which has Bayes relative risk on the x-axis and the "Bayesian p-value" for relative risk on the y-axis (Figure 2). The relative risk is the ratio of the risk of the AE in treatment versus control as described in the section on the Berry and Berry model. The "Bayesian p-value" of the relative risk is the posterior probability that the relative risk is less than or equal to one.

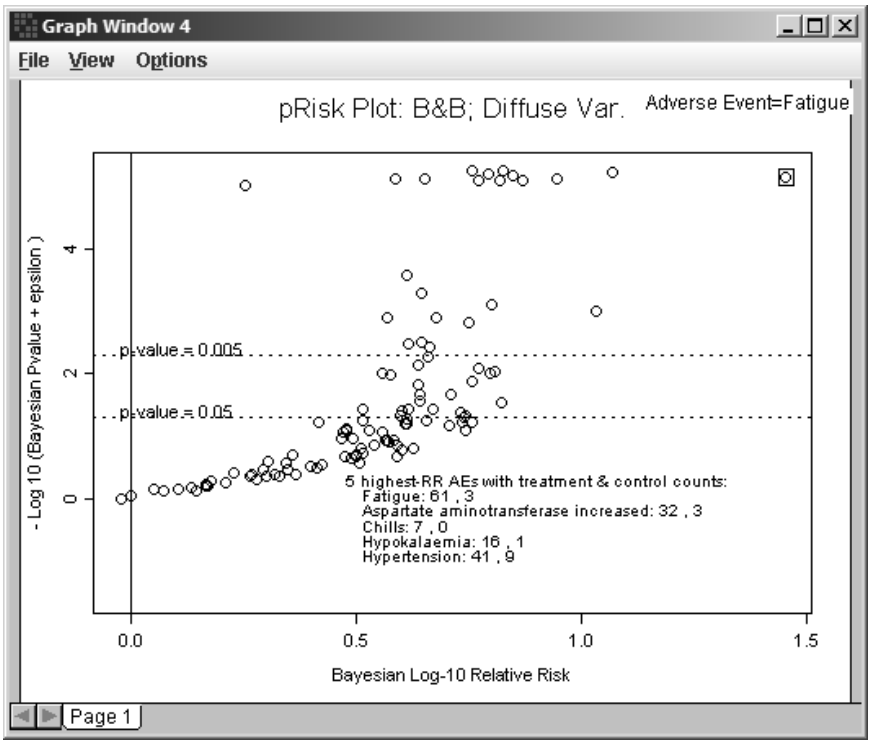


Figure 2. pRisk plot of adverse event preferred terms from the Berry and Berry (2004) analysis. Each point on the graph represents the log Bayes p-value versus log Bayes relative risk for a specific AE preferred term e.g. the point in the top right hand corner is the fatigue preferred term as shown above the graph as mouseover metadata. The graph in the figure is a screenshot from the S-PLUS java.graph() device, which supports metadata on mouseover and interactive drill-down in web browsers. This enables incorporation of the analysis into clinical / safety review applications such as Insightful Clinical review (see Figure 4 below).

In addition, the user can call

```
> bayesRR <- bayesRelativeRisk (aeSim = aeSimNotDiffuse,
               treatmentRiskParam = "theta.t",
              controlRiskParam = "theta.c", … )
# Pick the AEs with the largest estimated RR
> bayesRRbiggestEst <- restrictToBiggest(bayesRR, "rrEst", 20)
# Interval plot
> rrIntervalPlot (RR = bayesRRbiggestEst,
         intervalLabel = "Bayes Posterior Mean of Log-10 Relative Risk with 99% BCI",
         main = "A.E.s with the Largest Bayes Relative Risks")
```

which yields the following plot (Figure 3) showing Bayes relative risks and intervals for the AE preferred terms with largest Bayes relative risks. Note that usual (frequentist) relative risks can't be calculated for some preferred terms due to zero counts in the placebo arm.

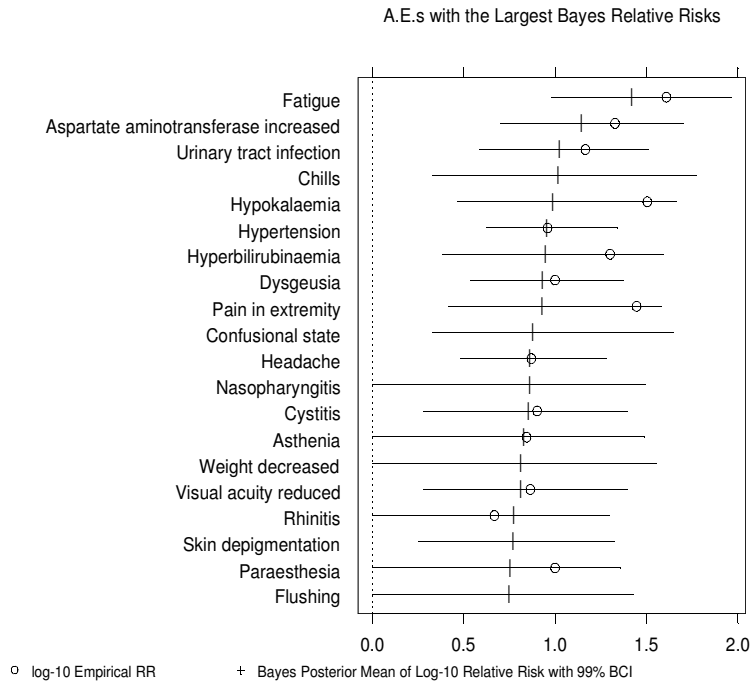


Figure 3. Bayes relative risk with credible interval, for the AE preferred terms with the highest Bayes relative risks. The graph shown in this figure is a WMF vector graph format can be resized in the MS Word document without affecting the graphic quality. The graph can also be sent to the S-PLUS java.graph() device, like Figure 1. This supports metadata on mouseover and interactive drill-down in web browsers. This enables incorporation of the analysis in to clinical / safety review applications such as Insightful Clinical review (see Figure 4 below).

BERRY AND BERRY (2004) ADVERSE EVENT ANALYSIS RESULTS

The Berry and Berry (2004) model provides a satisfying analysis of adverse events in a clinical trial. The model is structured in alignment with the desired interpretive structure i.e. bucketing the adverse events in to two groups – those affected by the treatment and those not affected by the treatment. This helps address the potential multiple comparisons issue, in a direct, interpretive manner.

The model also borrows strength among terms in the same SOC and among all the SOCs, thus addressing the sparse data issue. We studied the level of sensitivity to the choice of variance hyperparameters and the λ_α and λ_β hyperparameters, which govern the prior probability of no treatment effect for a preferred term within a system organ class. We found low sensitivity of the posterior distributions to the specifications of these hyperparameters.

The Berry and Berry hierarchical model has several levels, as given in the model description section. The first level is the specification of the risk parameter distributions; as described in that section, this level allows borrowing of

information within each system organ class. The second level is the specification of the random effects distributions; as discussed before, this level allows borrowing of information among the SOCs. We use a slightly simplified version of the Berry and Berry model, which maintains these levels and properties of the model while potentially improving identifiability.

DEPLOYMENT: BAYES ADVERSE EVENT ANALYSIS FOR CLINICAL DATA REVIEW

The graphs in Figures 2 and 3 succinctly summarize the results of the Bayes adverse event analysis using the Berry and Berry model. Figure 2 includes all the adverse events with Bayes relative risk on the x-axis and –log10(P-value) on the y-axis. Adverse event preferred terms in the upper right quadrant are potential safety signals with both high relative risk and low P-value for treatment v. placebo. Figure 3 shows the 20 adverse events with the highest Bayes relative risk. These two statistical graphics, summarizing results from the Bayes modeling, can be published from S-PLUS in a variety of formats including

- Vector graphics e.g. WMF, EMF, EPS for use in publications and presentations e.g. in MS Word and Power Point documents (e.g. S-PLUS wmf.graph() and postscript() devices)
- PDF graphics for use in publication reports (e.g. S-PLUS pdf.graph() device)
- SPJ graphics for use in interactive clinical review applications (e.g. S-PLUS java.graph() device)

We illustrate deployment of the Bayes modeling results using SPJ graphics for interactive clinical review. SPJ graphics (short for S-PLUS java) are created using the S-PLUS java.graph() device, which supports metadata on mouseover and interactive drill-down in web browsers. This enables incorporation of the Bayes analysis in to clinical / safety review applications such as Insightful Clinical Review Solution. Figure 4 below shows the pRisk plot from Figure 2 as part of the Insightful Clinical Review Solution.

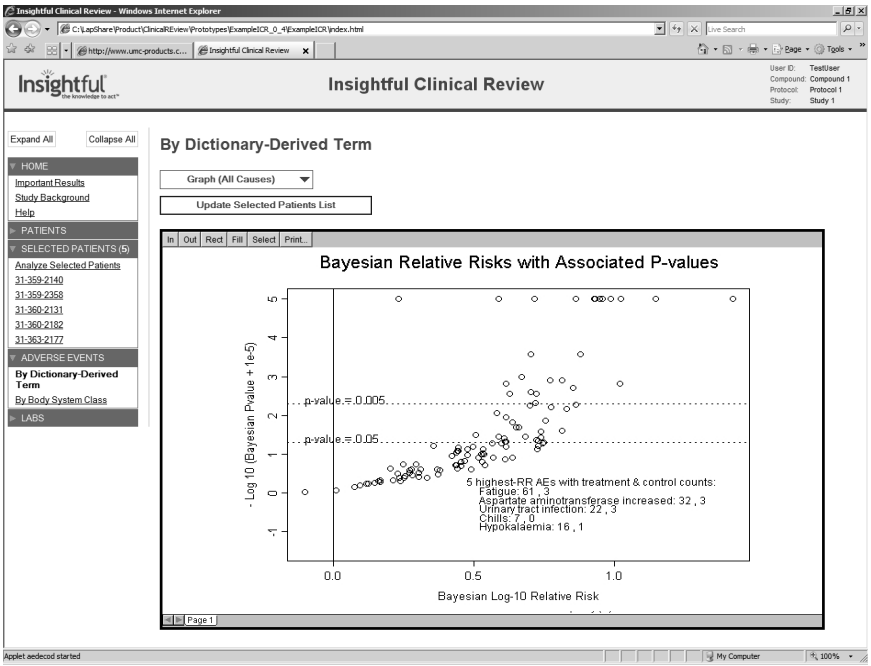
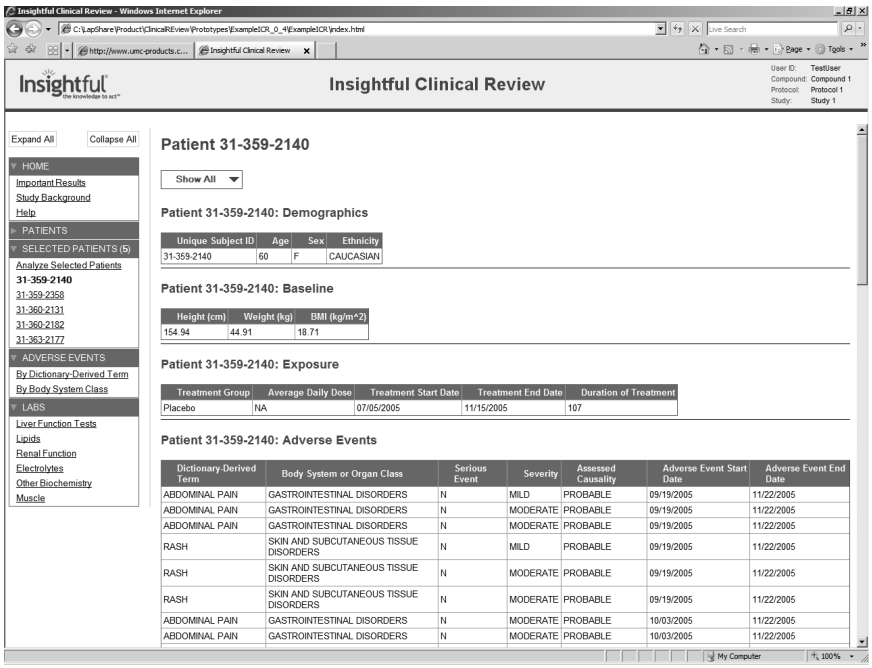


Figure 4. pRisk plot from Figure 2 as part of the Insightful Clinical Review Solution. The graph is obtained by clicking on the link 'By Dictionary Derived Term' in the Adverse Event section of the left-hand navigation tree. Each point on

the plot represents an adverse event preferred term.



The graph in Figure 4 is interactive; by clicking on a point and updating the selected for drill down in list of patients navigation the left updated. The panel is review the user can then for these patient profile subjects with a selected click. An single mouse patient profile excerpt of the selected for one such shown in patient is The patient Figure 5 below. profile excerpt shows tables of demographics, exposure and adverse events. The patient profile scrolls down to show labs, concomitant medications etc.

Figure 5. Drill down to patient profile from 'Selected Patients' list in left hand navigation tree. The workflow starts with the Bayes pRisk plot in Figure 4. The Berry and Berry Bayes model thus provides a guided review of the clinical trial data, enabling focused review of patients with potential treatment-emergent adverse events.

SUMMARY AND CONCLUSION

Bayesian methods are very useful in many applications across all phases of pharmaceutical research and development. The Berry and Berry (2004) model presented in this paper directly addresses the goals and challenges of adverse event analysis including data sparseness and treatment-emergence in the presence of multiple testing.

Results from the Berry and Berry model analysis are readily deployed as interactive clinical review graphics or publication reports as part of the Insightful Clinical Review and Reporting Solutions. Graphical summaries of the Bayes analysis published in the Insightful Clinical Review Solution enable a guided review of the clinical trial data; focusing attention on subjects who may have adverse events related to the treatment.

In this adverse events analysis example, S-PLUS, S+flexBayes and the Insightful Clinical Review Solution provide a collaborative solution for analysis and review of safety data. Statisticians and programmers may run the analysis; and clinicians can review the analysis results as part of their clinical review process. Programmers may also report results from within SAS and/or out to RTF, PDF or XML/HTML using the Insightful Reporting Solution.

The S+flexBayes package enables flexible Bayesian model-fitting within the collaborative S-PLUS environment. In particular, it provides accurate analysis, reporting and review of clinical / safety data. Examples include the analysis of adverse events as shown in this paper, Bayes adaptive trial design, population-pk modeling and joint safety-efficacy response modeling. The S+flexBayes implementation with S-PLUS, S+Server and the Insightful Clinical Solutions platform provides statisticians, programmers and clinicians with a flexible and productive solution for Bayes analysis, graphics, reporting and review.

REFERENCES

Berry, SM and Berry, DA (2004). "Accounting for multiplicities in assessing drug safety: a three-level hierarchical mixture model" Biometrics 60(2) 418-426.

Cook, SR, Gelman, A, and Rubin, DB (2006). "Validation of software for Bayesian models using posterior quantiles" Journal of Computational and Graphical Statistics 15, 675-692.

Gelman, A and Rubin, DB (1992). "Inference from iterative simulation using multiple sequences" Statistical Science 7, 457-511.

Geweke, J (1992). "Evaluating the accuracy of sampling-based approaches to calculating posterior moments" in *Bayesian Statistics 4* (ed JM Bernardo, JO Berger, AP Dawid, and AFM Smith). Clarendon Press, Oxford, UK.

Geweke, J (2004). "Getting it right: Joint distribution tests of posterior simulators" Journal of the American Statistical Association 99, 799-804.

Gilks, WR, Thomas, A and Spiegelhalter, DJ (1994). “A language and program for complex Bayesian modeling” The Statistician 43, 169-178.

Heidelberger, P and Welch, PD (1983). “Simulation run length control in the presence of an initial transient” Operations Research 31, 1109-1144.

O’Connell, M (2006). Statistical Graphics for the Design and Analysis of Clinical Development Studies. Insightful Webcast, July 2006.

Spiegelhalter, DJ, Abrams, KR and Myles, JP. (2004). Bayesian Approaches to Clinical Trials and Health-Care Evaluation. Wiley, NY.

ACKNOWLEDGMENTS

We would like to acknowledge Patrick Aboyoun, David Gohel and the Insightful Clinical Review team for the development of the iCR application shown in Figures 4 and 5.

RECOMMENDED READING

For an introduction to the use of Bayesian modeling in pharmaceutical applications, we recommend the end-user read the book, Bayesian Approaches to Clinical Trials and Health-Care Evaluation, by Spiegelhalter, DJ, Abrams, KR and Myles, JP.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Andrew Jack
Insightful Limited
Network House
Basing View
Basingstoke RG21 4HG
UK
Tel: +44 1256 339818
Email: firstname.surname@insightful.com (see above for name)
Web: http://www.insightful.com/uk

Brand and product names are trademarks of their respective companies.

APPENDIX: THE S+FLEXBAYES / BUGS BERRY AND BERRY FUNCTION

The `runBerryAndBerry()` function that is called in the S-PLUS session (see above) contains the following steps. These steps can be modified in many ways, illustrating the flexibility of the flexBayes engine.

First the model is specified for S+flexBayes using the BUGS syntax (Gilks, Thomas, and Spiegelhalter 1994). Since this syntax is very close to the S-PLUS language, the model is specified by defining a function `berryModel()`, containing code that defines the model. This function is never run; rather, the model specification is interpreted as text by the S+flexBayes package. The function for the Berry and Berry model is defined inside an outer wrapper function runBerryand Berry() below.

The runBerryAndBerry() function exposes some of the Berry and Berry model parameters, allowing sensitivity of model parameters to be easily studied in S-PLUS by simply assigning Bayes model runs (with different prior specifications) to different S-PLUS objects. The posterior output objects may then be studied in S-PLUS to assess sensitivity to priors. We found no sensitivity to the variance prior parameters (diffusePrior argument) or the pi parameter which sets the prior probability of treatment-emergence for an AE in a particular SOC.

```
runBerryAndBerry <- function(aeCounts, nIter = 10000, nThin = 1, diffusePrior = T, lam.alph =
0.0001)
{
  berryModel <- function (){
       for (i in 1:B) {
           for(j in 1:Nobs[i]) { ……
        logit(theta.t[i,j] )<- c[i,j] + a[i,j]
        c[i,j]~dnorm(mu.c[i], tau.c)
     }
```

After setting up the basic model parameters, the hyperparameters are specified:

```
     …… if (diffusePrior){
                 aeCounts$lambda.alpha <- 1
                 aeCounts$alpha.tau.a <- 0.5    ……
```

Next, a vector of initial values for the chain is specified:

```
     ……  for (bodySystem in 1:nBodySystems){
          nAEsThisBodySystem <- numAEsPerBodySystem[bodySystem]
                    cInit [bodySystem, 1:nAEsThisBodySystem] <- 0
          …… }
    initsBerry <- list(initsBerry)
```

Then a list is created of the parameters for which the posterior samples should be stored:

```
      …… parametersBerry <-
           c("s.tau.c", "s.rho.a", "s.rho.c", "theta.t", "theta.c", "tau.a",  "tau.c", "mu.a",
            "mu.c", "phi.a", "mean.a",,"mean.c", "psi.a", "noDrugEffect", "c", "a", "alpha.phi",
            "beta.phi")   ……
```

Finally, the model is run and the result is returned:

```
  aeSim <- posteriorSamples (data = aeCounts,
                   parametersToSave = parametersBerry,
                   model = berryModel, inits = initsBerry,
                   nChains = 1, nIter = nIter, nBurnin = 1000, nThin = nThin, DIC = F,
                   engine = engine)
  return(aeSim)
```