PhUSE 2008

Paper ST08

Statistically guided review of safety data in clinical trials

Michael O'Connell, Insightful Corporation, RTP, North Carolina, US
Harry Southworth, AstraZeneca, UK

ABSTRACT

This article and presentation outlines some approaches to statistically-guided review of safety data in clinical trials. The underlying statistical methods include an inside-out machine learning method for analysis of subject-level adverse event incidence data; a three-level hierarchical Bayesian mixture model for analysis of adverse event counts across subject populations, and an automated bias-reduced logistic regression method for analysis of individual adverse events on subject-level data. The statistical methods are implemented using the S-PLUS® libraries Forests, FlexBayes and BRLR. Results from these analyses are readily deployed as interactive graphics or publication reports using the S-PLUS Clinical Solutions framework. The interactive graphics link the summaries of the statistical analyses to subject-level data, including laboratory information and concomitant medications, from the clinical trial. This provides a statistically-guided review of drug safety in clinical trials.

INTRODUCTION

There has been much emphasis on statistical methods development for clinical trial design and analysis of efficacy endpoints. On the other hand, safety data are collected as concomitant information and historically have not been a focus for methodology development. This has recently started to change and there is considerable emerging interest in the statistical analysis of adverse event data (CIOMS, 2005; CDER, 2005). In addition, safety data are typically reported as tables and listings which are difficult to review and interpret.

This article and presentation outlines some approaches to safety data analysis and statistically-guided review of safety data in clinical trials. We briefly describe three statistical methods: an inside-out machine learning method for analysis of patient-level adverse event incidence data; a three-level hierarchical Bayesian mixture model for analysis of adverse event counts across subject populations, and an automated bias-reduced logistic regression method on individual adverse events using subject level data. These methods are described in more detail in Southworth and O'Connell (2009).

The methods are implemented in the S-PLUS environment of extensible packages and libraries. In conjunction with the collaborative S-PLUS Server, results from the safety analyses are readily deployed as interactive clinical review graphics or publication reports using the S-PLUS Clinical Solutions framework.

This paper is organized as follows: we first give a high-level introduction to adverse events analysis in the context of clinical drug development. We then describe the S-PLUS software environment and the S-PLUS packages/libraries supporting the statistical methods. We then describe the statistical methods and how relevant graphical summaries are used for statistically-guided review of safety data in clinical trials. We illustrate the complete workflow with an implementation of the hierarchical Bayesian mixture model, using the FlexBayes package.

SAFETY DATA ANALYSIS

In drug safety data analysis, one key goal is early detection of any increased risk of an adverse event under the clinical trial treatment. A wide range of possible adverse events are monitored in early- and late- phase drug trials, and any 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.

Adverse event data have historically been reported as simple tables and listings and have not been a focus for statistical methodology development. This has recently started to change and there is considerable emerging interest

in the statistical analysis of adverse event data (CIOMS, 2005; CDER, 2005).

SOFTWARE

The three safety data analysis methods we describe can be implemented using the S-PLUS libraries Forests, FlexBayes and BRLR. These libraries are available as packages that can be installed in to S-PLUS version 8. They are a few of the many packages and libraries that extend the functionality of core S-PLUS. With the release of S-PLUS 8, a freely-available repository of S-PLUS packages was created (http://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 framework of clinical solutions for pharmaceutical companies based on S-PLUS and S+Server, for statistical analysis and tabular/graphical reporting and review of clinical data. S-PLUS and S+Server run on multiple Windows and UNIX platforms, connect natively with a wide variety of data sources, and produce a rich 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.

The Forests library provides a very general collection of methods for fitting ensembles of trees. The underlying algorithm is based on recursive partitioning (Therneau and Atkinson, 1997), and Forests requires the arbor library (available in S-PLUS 8) to be loaded. The library includes functions that enable bagging (Breiman, 1996, Hastie et al., 2001), boosting (Friedman, 2002), random forests (Breiman, 2001) and boosted random trees. These are defined by argument selections to the primary `forests()` function. An alternative implementation of random forests, with a different set of features, may be obtained using the `randomForest` package available from CSAN.

The FlexBayes package includes its own internal MCMC engine for the class of hierarchical generalized linear models and connections to the open-source software WinBUGS and OpenBUGS engines. The 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. Once a model has been fit, a broad range of statistical inference can be performed by the user, utilizing the posterior samples obtained by FlexBayes. The FlexBayes package is currently in beta release and is available by sending email to beta@insightful.com. Alternative interfaces between S-PLUS and WinBUGS, with different feature sets, can be obtained from CSAN.

The BRLR package enables the fitting of bias-reduced logistic regression models by maximized penalized likelihood. This is based on a method outlined by Firth (1993). The package was originally created by Firth; the S-PLUS package was ported and is maintained by Southworth (2007) and is available on CSAN (http://csan.insightful.com).

Forests, FlexBayes and BRLR can be used to analyze different views on adverse events data. Results of such analyses can be summarized as graphics and tables that enable interactive medical monitoring and safety review. Such statistically-guided review enables focus on the adverse events that are most likely to be treatment related. This saves time and helps to focus and prioritize drug development in pharmaceutical companies.

STATISTICAL METHODS FOR ADVERSE EVENT ANALYSIS

MACHINE LEARNING ANALYSIS OF PATIENT-LEVEL ADVERSE EVENT INCIDENCE DATA

Our machine learning method for the analysis of patient-level adverse event incidence data is novel in that we turn the usual model formulation inside-out. In our formulation, the response variable is the treatment and the explanatory variables are the adverse events experienced by each subject. If the adverse events can be used to classify subjects to treatment groups with any degree of accuracy, it follows that there must be differences between the adverse event profiles of the treatments, and standard approaches can be used to establish where these differences exist.

To use this approach, the data set needs to be prepared so that the rows of the data are the subjects and the columns are the individual adverse events; the entries in the data matrix are patient-level incidences of the adverse events. There is one additional column for the treatment assignment to each patient. In some early-phase studies there may be more adverse event columns than subject rows. Our machine learning method works in all cases.

The S-PLUS Forests library includes a variety of tree ensemble methods including bagging, boosting, random forests and boosted random trees. All of these methods fit a forest of trees through bootstrapping, i.e. sampling with replacement, from the data matrix of subjects by adverse events as described above.

We use the patients left out of the bootstrap sample for each tree, i.e. the out-of-bag patients, to estimate prediction accuracy of the model and relative importance of the adverse events (the variable importance) in classifying the treatment. This enables a sorting of the adverse events with respect to their relatedness to treatment. We stratify our bootstrap sampling to choose samples in proportion to the number of subjects within each treatment. This is an important step for unbalanced designs as is often the case in clinical trials.

Other general purpose classifiers, including neural networks and support vector machines (Ripley, 1996) can also be used provided a measure of variable importance can be obtained. One advantage of the machine learning approach is that it analyzes data at the subject level so that interactions between adverse events can be used by the model. The Forest methods automatically model such interactions.

CODE ELEMENTS FOR MODEL FITTING

Fitting the forest models is simple with the S-PLUS Forests library, as illustrated below for bagging. The data are from a phase 2 clinical trial with approximately 150 subjects and 200 unique adverse events.

```
> library(Forests)
> aeBagOut <- forest( trt ~ ., data = aePatientLevel,
        nTrees = 200,
        boost = F,  #no boosting
        treeMethod = "class",
        nRandomSplitVars = 0) #bagging
```

The variable importances are then readily calculated from the model output as follows. A data frame is then prepared for plotting the variable importances for the top 24 adverse events.

```
> aeBagVI <- rfVariableImportanceSplitImprove(aeBagOut)
> aeBagVI <- aeBagVI [order(aeBagVI)]
> aeBagDf <- data.frame(pt.name=names(aeBagVI),varImp = aeBagVI)
```

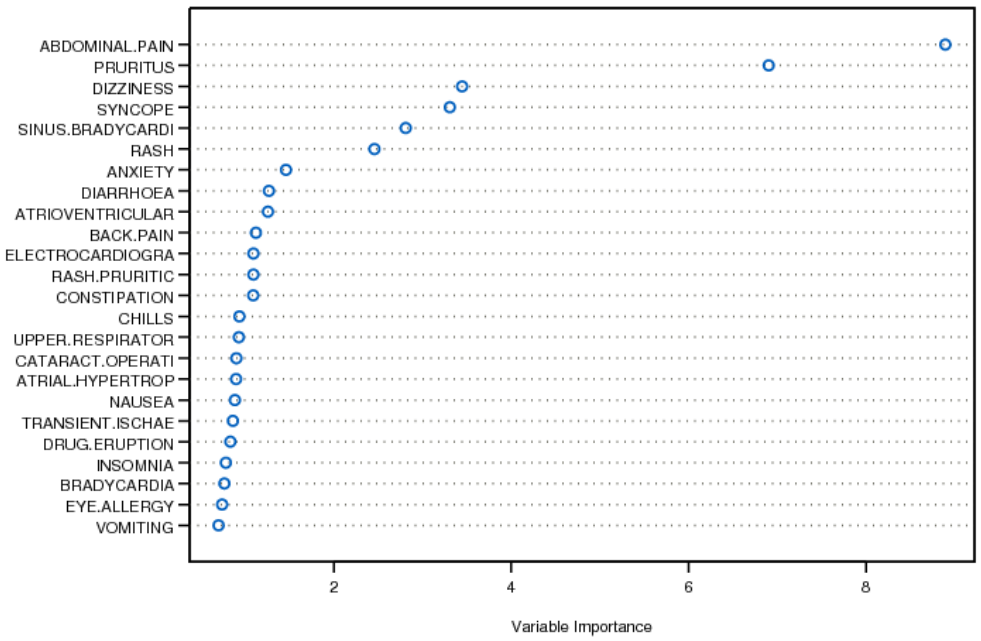

Figure 1. Dot plot of variable importances from bagging analysis of adverse event data from a clinical trial with approximately 150 subjects and 200 unique adverse events.

The Dotplot in Figure 1 was created using the Insightful Clinical Solutions, but can be similarly constructed using the function dotchart(). Note that there are six adverse events which stand out from the rest of the pack, namely abdominal pain, pruritis, dizziness, syncope, sinus brachycardi and rash.

HIERARCHICAL MODELING OF ADVERSE EVENT COUNT DATA

Berry and Berry (2004) describe a three-level hierarchical mixture model for analysis of adverse event counts. The counts represent totals of subjects experiencing the adverse events within treatments. 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 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 log-odds 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 an 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:

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

Based on our work, we have simplified the model, by relaxing σ_θ^2 to be non-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. This focuses 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

Preprocessing of the clinical trial data is first done to create a data object called `aeCounts`. This data object includes the counts of adverse events by treatment and the sample sizes. Next, the model is fit using the FlexBayes

function `runAEModel`:

```
> pr.notDiffPiSymm <- runAEModel(aeCounts, diffusePrior = F, piSymm = T,
engine = "OpenBUGS")
```

In this call, we have chosen to obtain 5000 samples (default) from the posterior distribution, after thinning by 10 to reduce autocorrelation of the samples. We have chosen to use the variance priors from Berry and Berry (2004), rather than more diffuse prior distributions. Similarly, we have chosen a symmetric prior distribution for the mixing parameter, π_i. 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. Other engines, e.g. WinBUGS, can be chosen by changing this argument setting. The resulting `pr.NotDiffPiSymm` object is an object of class `posterior`, which is defined and handled by a set of methods in the FlexBayes library.

Additional graphical summaries are available for using the S-PLUS Safety library, also available from Insightful. After loading that library, the user can call:

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

This yields a p-Risk plot (O'Connell, 2006), which has Bayes relative risk on the x-axis and the probability that the Bayes relative risk is less than one on the y-axis (Figures 2 and 3). The probability is obtained as a proportion of posterior samples where the relative risk is less than one. This is presented on the negative log10 scale so that adverse event terms in the upper right hand corner of the graph have both a high relative risk and a high confidence that the elevated relative risk is real.

Figure 2 is the usual WMF graph (windows metafile) as obtained by exporting the graph from the default graphsheet() graphics device in S-PLUS Windows, or by using the wmf.graph() device directly. WMF/EMF (enhanced metafile) graphs are vector format and can be resized inside Microsoft documents without affecting the graphic quality. Figure 3 is the same graph sent to the SPJ (S-PLUS java) graphics device. This graphics device supports metadata on mouseover and interactive drill-down in web browsers (see the Dizziness label shown in the top right hand corner as the mouse highlights the point with the largest relative risk on the graph). This enables incorporation of the analysis in to clinical / safety review applications such as the Insightful Clinical Solution framework (see Figure 4 below).

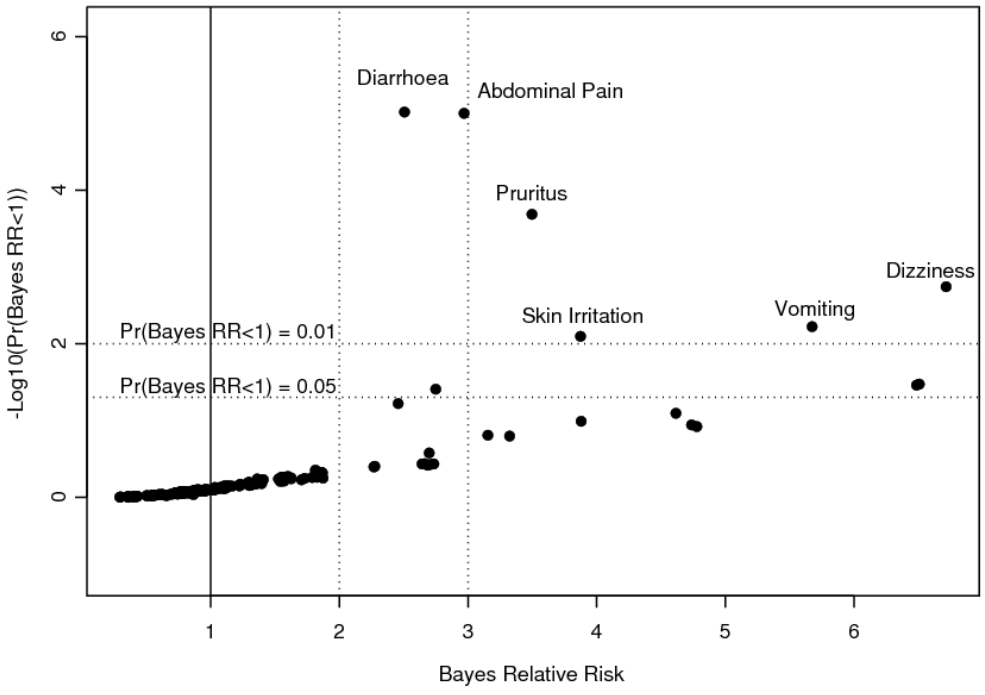

Figure 2. p-Risk plot of adverse event preferred terms from the hierarchical Bayes analysis of adverse event counts.

Each point on the graph represents the Bayes relative risk for a specific AE preferred term (x-axis) vs. the probability that the Bayes relative risk is less than one (y-axis). Adverse event terms in the upper right hand corner of the graph have both a high relative risk and a high confidence that the elevated relative risk is real.

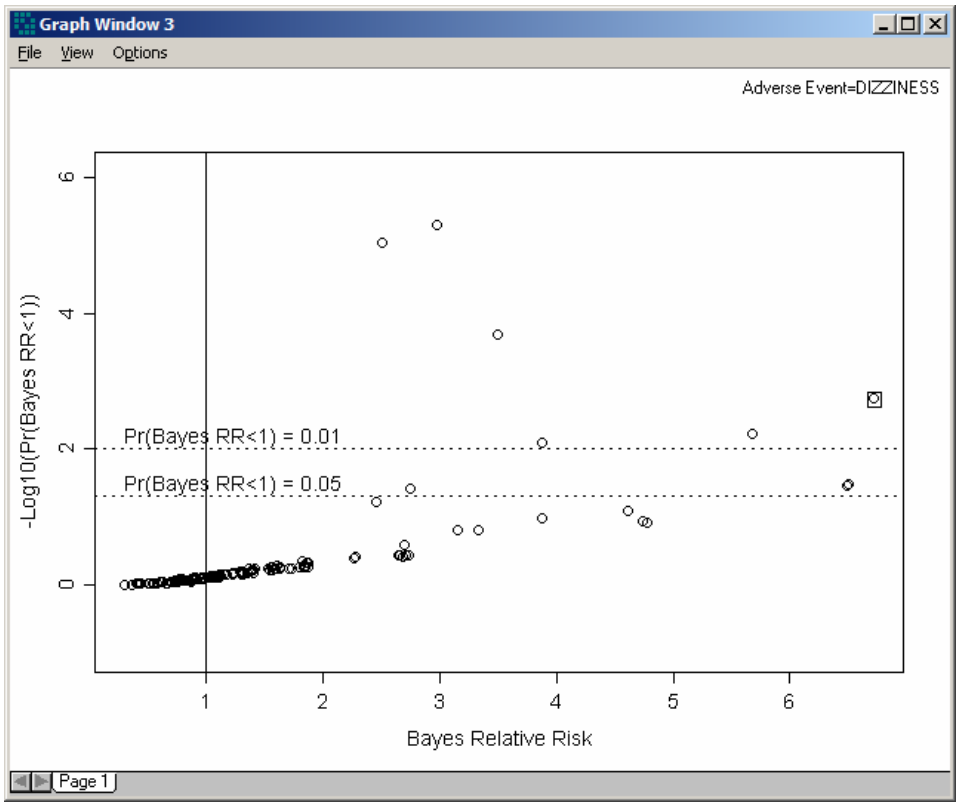

Figure 3. p-Risk plot of adverse event preferred terms from the hierarchical Bayes analysis of adverse event counts. This is a java.graph() version of Figure 2.

In addition, the user can call

```
> bayesRR <- bayesRelativeRisk (aeSim = pr.NotDiffPiSymm,
    treatmentRiskParam = "theta.t", controlRiskParam = "theta.c", … )
# Pick the AEs with the largest estimated RR
> bayesRRbiggestEst <- restrictToBiggest(bayesRR, "rrEst", 20)
# Interval plot for log10 Relative Risk
> rrIntervalPlot (RR = bayesRRbiggestEst,
    intervalLabel = "Bayes Posterior Mean of Relative Risk with 95%
    Credible Interval")
```

which yields the following plot (Figure 4) showing Bayes relative risks and intervals for the AE preferred terms with largest Bayes relative risks. Note that frequentist relative risks can't be calculated for some preferred terms due to zero counts in the placebo arm.

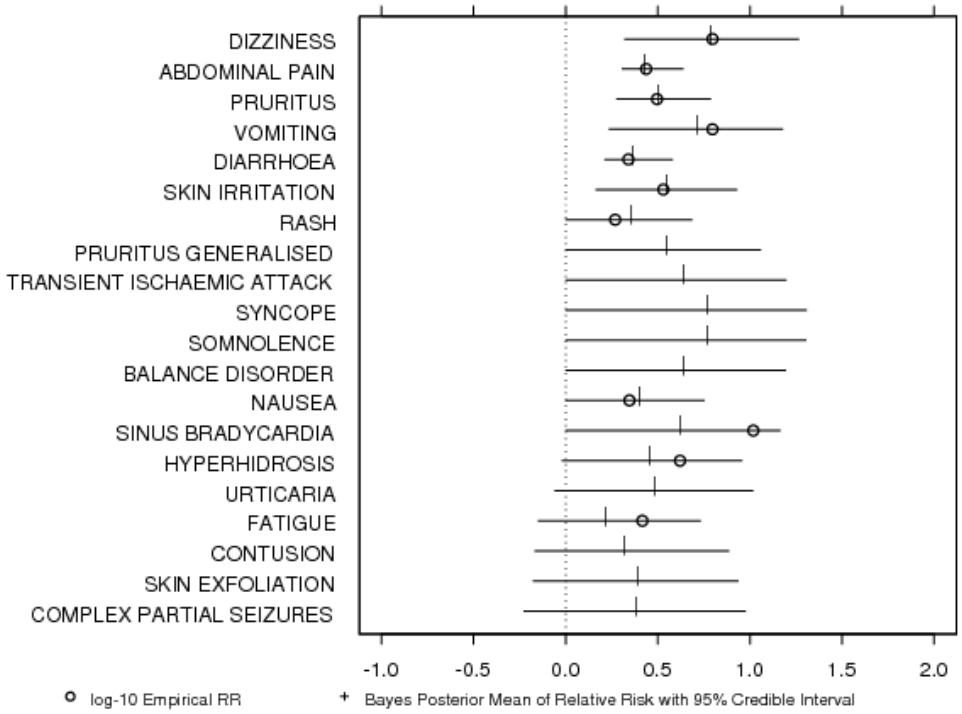

Figure 4. Bayes relative risk with credible interval, for the AE preferred terms with the highest Bayes relative risks. The relative risks are presented on the log10 scale on the x-axis.

Our modified Berry and Berry (2004) model fitted above provides an analysis of adverse event counts (by treatment) 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. 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.

Note that like the machine learning analysis, there are 6-7 adverse events that stand out from the rest. Four of these are in common with the machine learning analysis, namely dizziness, abdominal pain, pruritis and rash. The Bayes analysis identified an additional 3 terms, vomiting, diarrhea and skin irritation. The other terms identified in the machine learning analysis, syncope and sinus brachycardia, were shown to have long relative risk credible intervals that just touch 0 on the log10 scale (relative risk of 1) in the Bayes analysis. In a practical evaluation of these results, the top terms from all analyses would be noted and considered for follow-up evaluation.

PENALIZED LIKELIHOOD – AUTOMATED LOGISTIC REGRESSION

Our automated logistic regression approach leaves the adverse events as the response variable and uses treatment as an explanatory variable. We fit our models for each adverse event separately. For each adverse event we fit two models, one including treatment as an explanatory variable, one without treatment as an explanatory variable. We then assess the treatment effect for each adverse event by comparing the deviances from the two models. Since we are fitting a lot of models, we need an automated fitting method that is stable.

We use a penalized likelihood approach, corresponding to the saddlepoint approximation (Platt, 2000) or modified profile likelihood (Cox and Hinkley, 1986). The approach is also related to ridge regression and the lasso (Tibshirani, 1996). Our approach was derived by Firth (1993) in the context of exponential families for reducing the bias in parameter estimates. The Firth method reduces bias, leads to good interval estimates and always produces finite parameter estimates (Heinze and Schemper, 2002). Firth created an R package implementing the method; this was repurposed as the S-PLUS BRLR (bias reduced logistic regression) package by Southworth (2007) and is available for download on CSAN. An advantage of this approach over the machine learning and Bayesian approaches already

described is that effects of age, sex, body mass index and other covariates can be estimated. A disadvantage is that each model is treated as being independent of all others so that interactions between AEs are ignored.

CODE ELEMENTS FOR MODEL FITTING

Fitting the bias-reduced logistic regression models is simple with the S-PLUS BRLR package, as illustrated below. The data are from a phase 2 clinical trial with approximately 150 subjects and 200 unique adverse events.

```
> data(aeBRLR)
> aeNames = names(aeBRLR)[index]
> # Fit the full models
> pr.brlr = vector( mode = "list", length= length(aeNames) )
> for (i in 1:length(aeNames) ){
+     fo = formula( paste(aeNames[ i ], "~ age1 + age2 + age3 + age4 + SEX +
+     RACEN + BMIBL + TRTPCD" ) )
+     pr.brlr[[ i ]] = brlr( fo, data= aeBRLR )
> }

> # Refit without treatment
> pr.r.brlr = vector( mode = "list", length = length(aeNames) )
> for ( i in 1:length(aeNames) ) {
+    fo = formula( paste(aeNames[ i ], "~ age1 + age2 + age3 + age4 + SEX +
+    RACEN + BMIBL" ) )
+    pr.r.brlr[[ i ]] = brlr( fo, data= aeBRLR)
}
```

The difference in deviance (support) for the two models fit to each adverse event is then readily calculated from the model output as follows. A data frame is then prepared for plotting the support for the top 24 adverse events. In a practical setting, this whole process can be wrapped into a simple function.

```
# Graph the VIs
> pr.brlr.vi = sapply( 1:length(aeNames), function( i, a, b )
        a[[ i ]]$penalized.deviance - b[[ i ]]$penalized.deviance,
        a = pr.r.brlr, b = pr.brlr )
> names(pr.brlr.vi) = aeNames
# plot of change in penalized deviance
> dotchart(rev(sort(pr.brlr.vi)))
```

The Dotplot in Figure 5 was created using the Insightful Clinical Solutions, but can be similarly constructed using the function dotchart(). Note that there are three adverse events which stand out from the rest of the pack, namely abdominal pain, pruritis and dizziness, syncope. These three events were also identified using the machine learning and Bayes methods. In a practical evaluation of these results, the top terms from all analyses would be noted and considered for follow-up evaluation.

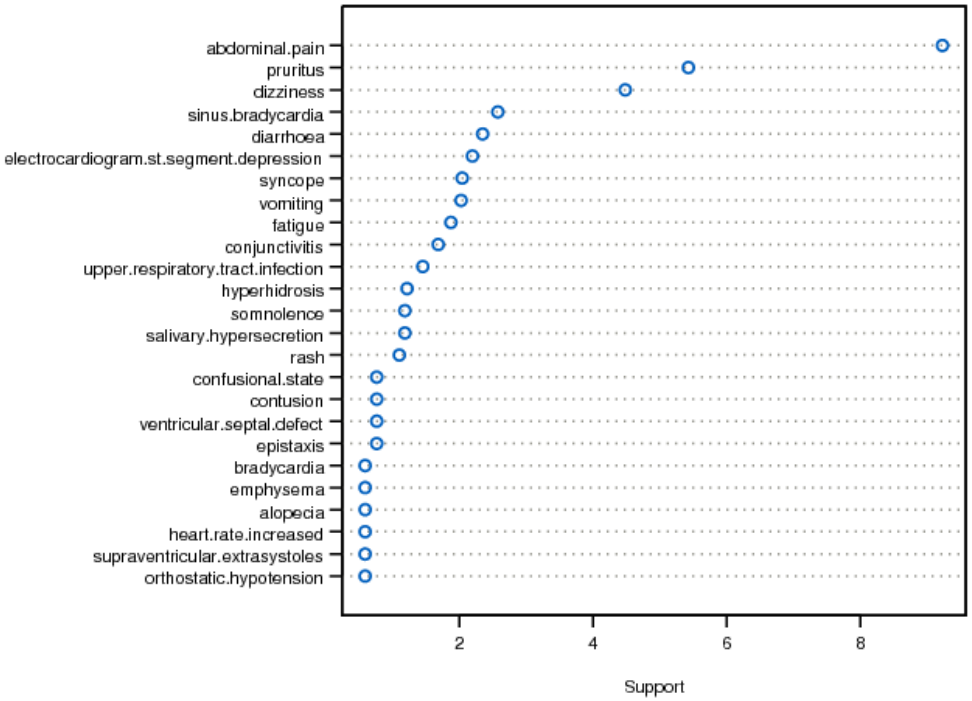

Figure 5. Dot plot of support (change in deviance for models with and without the treatment effect) from bias-reduced logistic regression of adverse event data from a clinical trial with approximately 150 subjects and 200 unique adverse events.

DEPLOYMENT: ADVERSE EVENT ANALYSIS FOR CLINICAL DATA REVIEW

We illustrate the statistically-guided clinical review process using the hierarchical Bayes analysis of adverse event counts analysis described above. The graphs in Figures 2-4 summarize the results of this analysis. Figures 2 and 3 include all the adverse events with Bayes relative risk on the x-axis vs. the probability that the Bayes relative risk is less than one on the y-axis. Adverse event terms in the upper right hand corner of the graph have both a high relative risk and a high confidence that the elevated relative risk is real. Figure 2 shows the graph in WMF format, and Figure 3 in (interactive) SPJ format. Different formats of graphics are useful for different reporting applications e.g.

- Vector graphics e.g. WMF, EMF, EPS are used in publications and presentations e.g. in MS Word and Power Point documents. The S-PLUS graphsheet and the wmf.graph() and postscript() devices are useful for creating graphs of this type.
- PDF graphics for use in publication reports; the S-PLUS pdf.graph() device is used for graphs of this type.
- SPJ graphics for use in interactive clinical review applications; the S-PLUS java.graph() device is used for graphs of this type.

We illustrate deployment of the adverse event modeling results using SPJ graphics for interactive clinical review. SPJ graphics 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 analysis in to clinical / safety review applications such as S-PLUS Clinical Solution framework. Figure 6 below shows the p-Risk plot from Figure 3 as part of this framework.

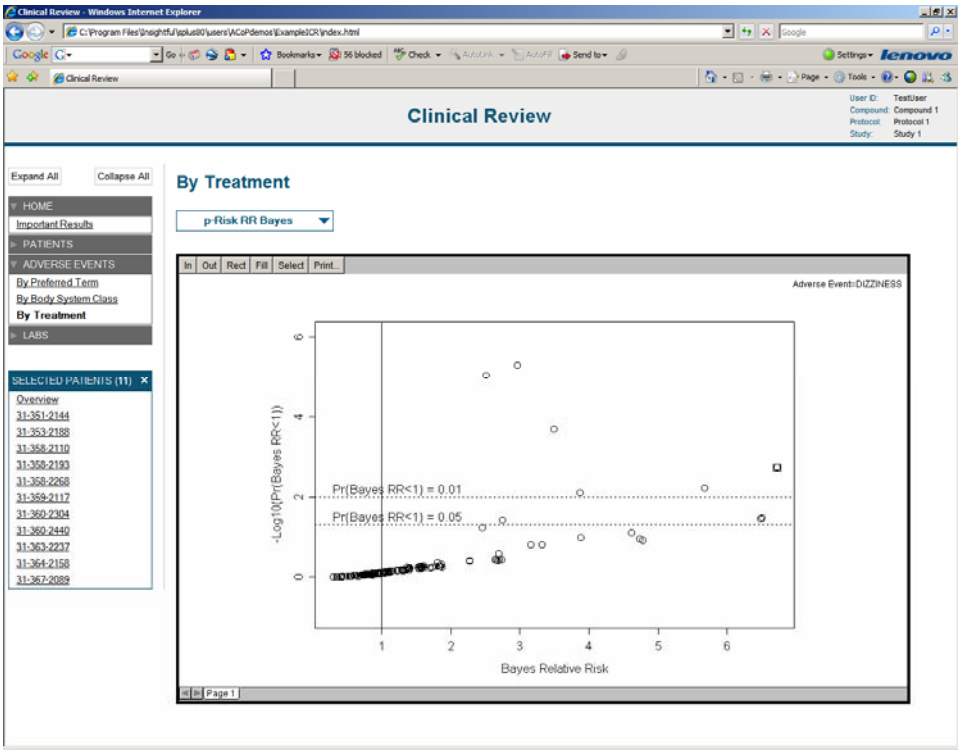

Figure 6. p-Risk plot from Figure 3 as part of the S-PLUS Clinical Solutions framework. The graph is obtained by clicking on the link 'By Treatment' in the Adverse Event section of the left-hand navigation tree. Each point on the plot represents an adverse event preferred term.

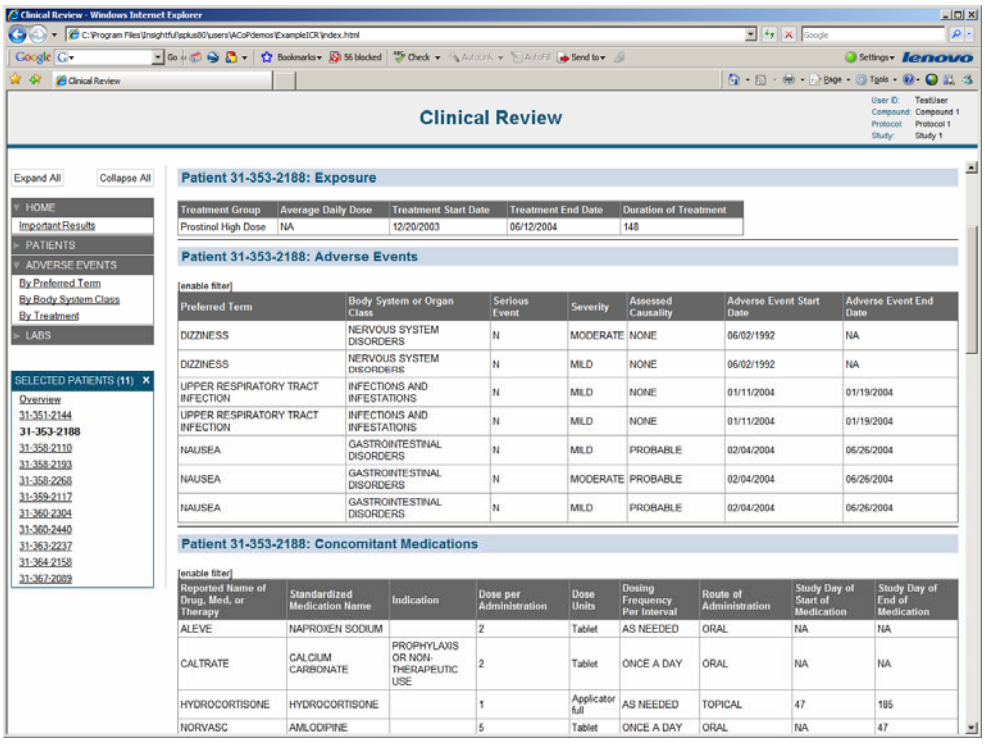

Figure 7. Drill down to patient profile from 'Selected Patients' list in left hand navigation tree. This drill-down from Figure 6 to Figure 7 provides a guided review of the clinical trial data, enabling focused review of patients with potential treatment-emergent adverse events.

The screen shots in Figures 6 and 7 are interactive web pages; by clicking on a point (e.g. Dizziness in Figure 6), the selected patient list is updated to include all subjects experiencing the Dizziness adverse event. The user can then review the patient profile for these selected subjects with a mouse click. An excerpt of the patient profile for one such selected subject is shown in Figure 7. The patient profile excerpt shows tables of adverse events and concomitant medications. The patient profile scrolls down to show labs, vitals, medical history etc. These web pages can be linked from other clinical review applications such as Spotfire, as shown in Figure 8.

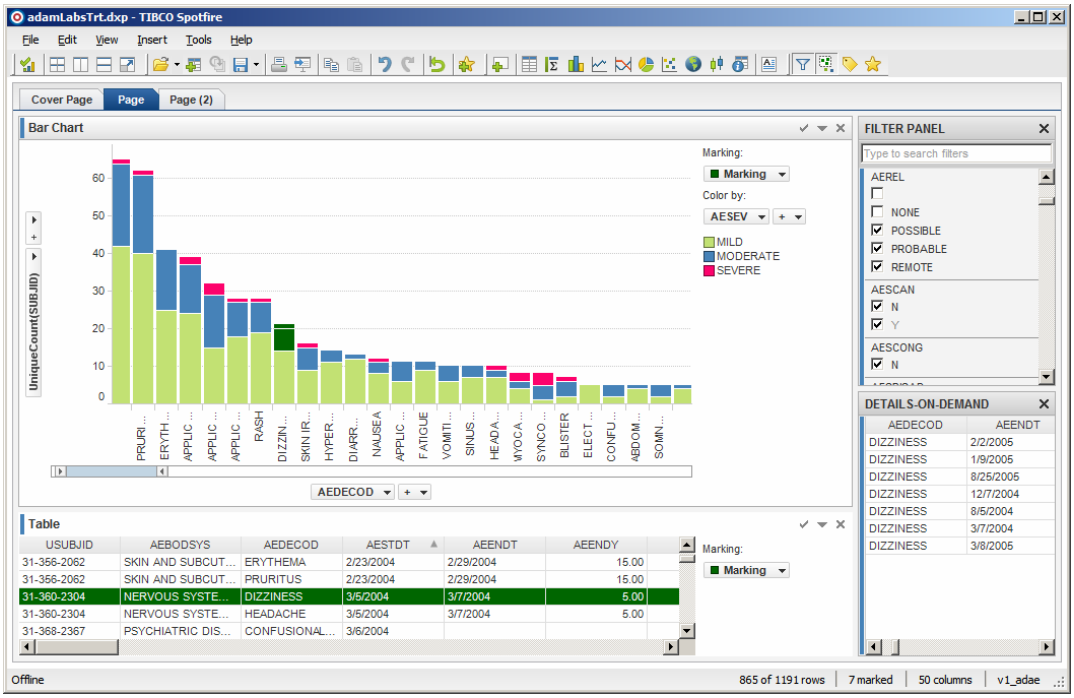

Figure 8. Clinical data (adverse events) from Figures 6-7 in the interactive visualization application, Spotfire. Spotfire views on clinical data can be saved to a web-player and mashed-up with the graphical and tabular presentations from the S-PLUS Clinical Solutions framework shown in Figures 6-7.

SUMMARY AND CONCLUSION

We have outlined an approach to statistically-guided review of safety data in clinical trials. This includes statistical modeling, succinct graphical summaries of the modeling, and population-patient drill-down where detailed review of subjects is undertaken for adverse events shown to be potentially treatment-emergent from the statistical analyses.

The underlying statistical methods include an inside-out machine learning method for analysis of patient-level adverse event incidence data; a three-level hierarchical Bayesian mixture model for analysis of adverse event counts across subject populations, and an automated bias-reduced logistic regression method for analysis of individual adverse events on patient-level data. The statistical methods are implemented using the S-PLUS® packages/libraries Forests, FlexBayes and BRLR. The methods produce similar results. The machine learning method uses the finest scale subject-level data in a single analysis. The logistic regression method analyses each adverse event in turn at the subject level. The hierarchical Bayes method uses count data across treatments.

Results from any of the statistical analyses are readily deployed as interactive graphics or publication reports using the S-PLUS Clinical Solutions framework. The interactive graphics link to graphical and tabular summaries of the subject-level data, including laboratory information and concomitant medications, from the clinical trial. The graphical and tabular summaries may also be linked to other web-based applications including workflows published from the interactive visualization application, Spotfire. These workflows provide a statistically-guided review of drug safety in clinical trials.

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.

Breiman, L. (1996). Bagging predictors. M*achine learning* 24, 123-140.

Breiman, L. (2001). Random Forests. M*achine learning* 45, 5-32.

CDER (Center for Drug Evaluation and Research), US HHS FDA (2005). Reviewer Guidance: Conducting a clinical safety review of a new product application and preparing a report on the review

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.

Cox, D and Hinkley, DV. (1986). *Theoretical Statistics*. Chapman and Hall.

CIOMS (Council for International Organizations of Medical Sciences), Working Group VI (2005). Management of Safety Information from Clinical Trials.

Firth, D (1993). Bias reduction of maximum likelihood estimates, *Biometrika* 1980, 27-38.

Friedman, JH (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38, 367-378.

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

Hastie, T, Tibshirani, R and Friedman, JH (2001). *The Elements of Statistical Learning*. Springer.

Heinze, G and Schemper, M (2002). A solution to the problem of monotone likelihood in logistic regression. *Statistics in Medicine*, 21, 2409-2419.

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

Platt, RW (2000). Saddlepoint approximations for small sample logistic regression problems, *Statistics in Medicine* 19, 323-334.

Ripley, BD (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.

Southworth, H and O'Connell, M (2009). Statistical Data Mining for the Analysis of Adverse Event data in Clinical Trials. Submitted to *Biostatistics*.

Southworth, H (2007). The S-PLUS BRLR package. CSAN: http://csan.insightful.com

Tibshirani, R (1996). Regression shrinkage adn selection via the lasso. *Journal of the Royal Statistical Society Series B*, 58, 267-288.

Therneau, T and Atkinson, EJ (1997). An introduction to recursive partitioning using the RPART routines. Mayo Foundation Technical Report http://mayoresearch.mayo.edu/mayo/research/biostat/upload/61.pdf.

CONTACT INFORMATION

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

Michael O'Connell
Email: moconnell@insightful.com

Harry Southworth
Email: harry.southworth@astrazeneca.com