PharmaSUG China 2026 - Paper XX-###

Implementing a Single-Agent Bayesian Logistic Regression Model in SAS for Dose Escalation in Phase I Oncology Trials

Zhanwen Peng, Akeso, Inc.

ABSTRACT

The Bayesian logistic regression model (BLRM) is a widely used model-based design for dose escalation in phase I oncology trials; nevertheless, reusable SAS implementations that make MCMC checking explicit are seldom reported in conference papers. This paper describes a pragmatic SAS workflow for a single-agent BLRM. The BLRM models dose-limiting toxicity (DLT) counts using a binomial likelihood and a log-dose logistic regression, assigns a bivariate normal prior to transformed model parameters, and applies escalation with overdose control (EWOC) to support dose recommendation. The implementation is structured as modular SAS macros for single-chain posterior sampling, multi-chain formal posterior inference, posterior toxicity summarization, EWOC-based decision support, single-trial execution, repeated simulation, and simulation summarization. In a formal multi-chain analysis, the wrapper runs independently initialized PROC MCMC chains, retains within-chain effective sample size (ESS), Monte Carlo standard error (MCSE), autocorrelation, Geweke, and trace-review outputs for every chain, and calculates the classical Gelman-Rubin R-hat as a between-chain diagnostic. Pooled posterior draws are used for formal toxicity summaries and EWOC decision support only after both levels of diagnostic review are satisfactory. This separation retains a computationally efficient single-chain engine for simulation while providing a transparent, auditable, and reusable BLRM workflow for formal interim analysis.

Keywords: BLRM, dose escalation, EWOC, PROC MCMC, MCMC diagnostics, R-hat, simulation

INTRODUCTION

Phase I dose-escalation trials are designed to identify dose levels with acceptable toxicity while protecting patient safety during early clinical development. In oncology trials, this objective has traditionally been framed in terms of identifying a maximum tolerated dose (MTD), although many modern development programs ultimately determine a recommended phase II dose (RP2D) using toxicity together with pharmacokinetic, pharmacodynamic, and preliminary activity information. Even when the final recommendation is not defined purely by toxicity, reliable characterization of the dose-toxicity relationship remains central to phase I decision-making.

Traditional rule-based designs such as the 3+3 design remain familiar because of their simplicity, but they make limited use of accumulated data and do not explicitly model the underlying dose-toxicity curve. Model-based approaches address this limitation by borrowing information across dose levels and translating observed dose-limiting toxicity data into posterior distributions for toxicity risk. Among these methods, the Bayesian logistic regression model (BLRM) has become a prominent option in oncology because it represents toxicity on a continuous dose scale, allows formal incorporation of prior information, and supports probabilistic dose-escalation decisions through escalation with overdose control (EWOC).

In practice, study teams need more than the mathematical definition of BLRM. They need a reproducible programming workflow that can update the model at each interim review, summarize posterior toxicity at observed and candidate doses, apply EWOC-based decision rules consistently, and support repeated simulation under alternative truth scenarios. In organizations where SAS remains the primary environment for statistical programming, validation, and reporting, a transparent SAS implementation is

especially useful. The purpose of this paper is therefore not to propose a new design, but to present a practical and reusable SAS workflow for single-agent BLRM dose escalation.

Because posterior decision quantities are approximated by finite MCMC draws, an operational implementation should distinguish model specification from sampling adequacy. For a formal interim analysis, each independently initialized chain must demonstrate adequate within-chain behavior, and the chains must also agree with one another before retained draws are pooled. These numerical checks provide evidence that the draws adequately represent the specified posterior; they do not validate the clinical assumptions or convert a toxicity-based recommendation into an RP2D decision.

Publicly accessible implementation-focused write-ups of BLRM in SAS appear to remain limited. Available public materials include a PSI 2018 poster describing a user-friendly SAS implementation built around PROC MCMC and EWOC decision support, whereas publicly available trial documents often indicate that Bayesian BLRM modeling was performed in R with WinBUGS even when SAS was used for routine analyses and reporting. By contrast, public software support for BLRM is more visible in R through packages such as OncoBayes2 and decider.

SINGLE-AGENT BLRM METHODOLOGICAL FRAMEWORK

MODEL FORMULATION

For a dose level d, let n(d) denote the number of treated subjects and r(d) the number with DLT. The single-agent BLRM begins with a binomial model for the observed DLT count and then links the toxicity probability to dose on the log-dose scale. In implementation, posterior sampling is performed on transformed parameters so that positivity constraints are respected and numerical stability is improved in PROC MCMC.

$$r(d) \mid n(d) \sim Binomial(n(d), \pi(d))$$

$$logit(\pi(d)) = log(\alpha) + \beta\, log(d/d^*)$$

$$\theta = (log\ \alpha, log\ \beta) \sim N(m, S)$$

Here d* is a prespecified reference dose used to normalize the dose range. Under this parameterization, α represents the DLT odds at the reference dose and β governs the steepness of the dose-toxicity relationship on the log-dose scale. The transformed parameter vector θ is assigned a bivariate normal prior with mean vector m and covariance matrix S.

DESIGN INPUTS AND DECISION RULE

Before trial initiation, the user specifies the planned dose grid, the reference dose, prior parameters, the starting dose, the underdosing, target-toxicity, and overdosing intervals, the EWOC bound, the cohort size, the maximum sample size, the dose-skipping convention, and the stopping rule. After each cohort completes DLT assessment, the posterior distribution is updated from the cumulative dose-level counts and posterior region probabilities are evaluated for each candidate dose.

Among doses that satisfy the overdosing constraint imposed by EWOC, the next recommended dose is typically the one with the greatest posterior probability of lying in the target-toxicity interval, while still respecting any prespecified restriction on dose skipping.

The prior specification and the EWOC threshold are design inputs rather than default settings. In practice, they should be prespecified and then examined by simulation before the design is used operationally. This combination of model-based updating and operational calibration makes BLRM attractive in settings where dose-escalation decisions must be justified to both clinicians and statisticians.

SAS IMPLEMENTATION

OVERVIEW OF THE INTERIM WORKFLOW

The single-agent implementation is organized around a two-tier MCMC workflow that separates single-chain computation from formal posterior inference. The base macro, blrm_main, runs one PROC MCMC chain and can return within-chain ESS, MCSE, autocorrelation, Geweke, and optional trace and autocorrelation plots. It can be used independently for development or targeted inspection, and it is also the computational engine invoked internally by the multi-chain wrapper.

For an operational interim review, the recommended sequence is blrm_main_mc, within-chain diagnostic review, between-chain diagnostic review, decision-stability review, blrm_stat, and blrm_ewoc. The multi-chain wrapper runs independent blrm_main chains, retains the within-chain diagnostic output for every chain, pools retained draws, and creates a classical Gelman-Rubin R-hat output for log_alpha and log_beta. Thus, a separate extra single-chain analysis is not required solely to obtain single-chain diagnostics: those diagnostics are already produced for each internal chain of blrm_main_mc.

REPRESENTATIVE ROLES OF THE CORE MACROS

blrm_main fits the model to the cumulative dose-level dataset, which must contain one record per observed dose with variables dose, n, and dltn. It returns posterior samples for the transformed model parameters. The optional diagnostics=1 setting retains within-chain ESS, MCSE, autocorrelation, and Geweke output, and diagplots=1 can display trace and autocorrelation plots for targeted inspection. When blrm_main is invoked by blrm_main_mc, the same within-chain diagnostics are generated for every formal chain.

blrm_main_mc is the wrapper recommended for formal interim inference. It runs multiple independently initialized chains, appends chain identifiers to the retained draws, collects the within-chain diagnostic tables from each chain, pools the retained draws, and calculates a classical Gelman-Rubin R-hat for log_alpha and log_beta. The pooled output should be supplied to blrm_stat and blrm_ewoc only after both within-chain and between-chain diagnostics have been reviewed.

blrm_ewoc classifies each candidate dose into underdosing, target, and overdosing regions and identifies which doses satisfy the prespecified EWOC bound. Its output is a toxicity-based decision-support table; protocol-specific operational constraints and cross-functional RP2D considerations remain outside this macro.

WORKED EXAMPLE

A simple interim example illustrates practical use of the macros. Five candidate doses are considered: 1, 5, 10, 12, and 15 mg, with 10 mg used as the reference dose. At the time of review, three subjects have been treated at 1 mg with no observed DLTs, and six subjects have been treated at 5 mg with one observed DLT. This compact example is sufficient to show the full path from cumulative trial data to posterior summaries and EWOC-based decision support.

Table 1 shows the cumulative DLT data used in the worked example.

Dose (mg)	N Treated	N with DLT
1	3	0
5	6	1

Table 1. Cumulative DLT Data Used in the Worked Example

Display 1 shows the formal multi-chain SAS workflow for the worked example. The call creates diagnostic outputs for each internal chain and a between-chain R-hat table, then uses the pooled posterior draw dataset for posterior toxicity summarization and EWOC classification. Display 2 shows the corresponding diagnostic-review code.

```
%blrm_main_mc(
 datain=dlt_dummy,
 dataout=out_mcmc_mc,
 ref_dose=10,
 mu1=-4, mu2=0.5, v1=4, v2=1, rho=0,
 alpha=0.05, nbi=2000, nmc=10000, nchain=4,
 seed=9527, diagnostics=1,
 chain_diagout=out_mcmc_mc_chain_diag,
 diagplots=0, ac_lags=%str(1 5 10 50),
 init_random=1, rhat_cutoff=1.01,
 postsumout=out_mcmc_mc_postsum,
 diagout=out_mcmc_mc_rhat, debug=0
);

%blrm_stat(
 dose_list=%str(1,5,10,12,15),
 ref_dose=10, mcmcout=out_mcmc_mc,
 dataout=d1, debug=0,
 stat=%str(mean, std, q1, median, q3)
);

%blrm_ewoc(
 dose_list=%str(1,5,10,12,15),
 ref_dose=10, mcmcout=out_mcmc_mc,
 dataout=ewoc_next,
 ud=0.16, od=0.33, ewoc=0.25, debug=0
);
```

Display 1. Formal Multi-Chain Posterior Inference, Toxicity Summarization, and EWOC Classification

Table 2 reports posterior toxicity summaries across the candidate doses.

Dose (mg)	Posterior Mean	SD	Q1	Median	Q3
1	0.02147	0.03490	0.00166	0.00780	0.02508
5	0.07645	0.07818	0.02003	0.04991	0.10631
10	0.16410	0.15652	0.04864	0.11262	0.23290
12	0.19830	0.18678	0.05892	0.13747	0.28471
15	0.24356	0.22241	0.07231	0.17046	0.35434

Table 2. Posterior Toxicity Summary by Candidate Dose

Table 3 shows the corresponding EWOC-based dose classification.

Dose (mg)	Pr(Underdosing)	Pr(Target)	Pr(Overdosing)	EWOC Acceptable
1	0.9881	0.0116	0.0003	True
5	0.8629	0.1223	0.0148	True
10	0.6232	0.2384	0.1384	True
12	0.5509	0.2523	0.1969	True
15	0.4779	0.2481	0.2740	False

Table 3. EWOC-Based Dose Classification from the Worked Example

In this example, 10 mg and 12 mg both satisfy the EWOC criterion and have larger target-region probabilities than the lower doses, whereas 15 mg exceeds the overdosing bound and is not considered acceptable under the prespecified rule. In the formal workflow shown in Display 1, Table 2 and Table 3 are generated from pooled posterior draws only after diagnostic review. The model-selected dose is a toxicity-based recommendation under the prespecified rule; selection of an RP2D remains a broader cross-functional decision.

MCMC DIAGNOSTICS AND FORMAL INTERIM ANALYSIS

The finite posterior sample produced by MCMC must be adequate for the decision quantities derived from it. The diagnostics edition therefore treats within-chain and between-chain checks as complementary requirements. blrm_main_mc is not an alternative to single-chain diagnostics: it runs blrm_main independently for each chain and preserves the ESS, MCSE, autocorrelation, and Geweke outputs produced within every chain, while also calculating a classical Gelman-Rubin R-hat across chains for log_alpha and log_beta.

The default formal workflow uses four chains with independent random initialization. Retained draws include chain, chain_seed, and chain_iteration so that trace behavior can be reviewed separately for each chain and compared across chains. Within-chain review assesses whether a given chain has reached stable behavior and produces sufficiently precise Monte Carlo estimates; between-chain review assesses whether independent chains are exploring the same posterior distribution. The macro reports R-hat only for the two sampled model parameters and it is unavailable when fewer than two chains are used. The rhat_cutoff parameter defaults to 1.01 and acts as a user-specified operational flag rather than a standalone certificate of convergence.

A complete review considers trace behavior, autocorrelation, ESS, MCSE, and Geweke output for each chain; visual agreement of chain trajectories and R-hat are then used to assess between-chain agreement. The final check is decision stability: posterior underdosing, target-toxicity, and overdosing probabilities, especially for doses close to the EWOC boundary or to one another in target probability, should be insensitive to ordinary Monte Carlo variation. ESS and MCSE should therefore be interpreted relative to the precision needed to distinguish candidate dose decisions. These diagnostics assess sampling adequacy for the specified model; they do not validate the prior, the dose-toxicity model, or the broader clinical appropriateness of the eventual RP2D.

Display 2 shows representative output-review code for both diagnostic levels. Table 4 catalogs the diagnostic datasets returned by the wrapper. Numerical diagnostic values are intentionally not presented

as universal benchmarks because they depend on the accumulated trial data, prior, sampler settings, and retained run.

```
/* Within-chain diagnostics, retaining chain identifiers */
proc print data=out_mcmc_mc_chain_diag_ess noobs; run;
proc print data=out_mcmc_mc_chain_diag_mcse noobs; run;
proc print data=out_mcmc_mc_chain_diag_autocorr noobs; run;
proc print data=out_mcmc_mc_chain_diag_geweke noobs; run;

/* Between-chain diagnostic */
proc print data=out_mcmc_mc_rhat noobs; run;

ods graphics on;
/* Trace review: each line is one independent chain */
proc sgplot data=out_mcmc_mc;
 series x=chain_iteration y=log_alpha / group=chain;
 xaxis label="Retained iteration within chain";
 yaxis label="log_alpha";
run;
proc sgplot data=out_mcmc_mc;
 series x=chain_iteration y=log_beta / group=chain;
 xaxis label="Retained iteration within chain";
 yaxis label="log_beta";
run;
```

Display 2. Representative Review of Within-Chain and Between-Chain MCMC Diagnostics

Diagnostic Scope	Output	Description
Pooled draws	out_mcmc_mc	Pooled draws with chain labels; used for trace review and downstream summaries.
Within-chain	out_mcmc_mc_chain_diag_ess	Effective sample size, by chain.
Within-chain	out_mcmc_mc_chain_diag_mcse	Monte Carlo standard error, by chain.
Within-chain	out_mcmc_mc_chain_diag_autocorr	Autocorrelation at requested lags, by chain.
Within-chain	out_mcmc_mc_chain_diag_geweke	Geweke diagnostic output, by chain.
Between-chain	out_mcmc_mc_rhat	R-hat, cutoff, convergence flag, and variance components.
Pooled summary	out_mcmc_mc_postsum	Pooled posterior summary for log_alpha and log_beta.

Table 4. Diagnostic Outputs from blrm_main_mc

Between-Chain Diagnostic Review: Gelman-Rubin R-hat

Posterior Parameter	Number of Chains	Minimum Draws per Chain	Maximum Draws per Chain	Within-Chain Variance	Variance of Chain Means	Rhat Diagnostic Status	Rhat Convergence Flag	Rhat Convergence Cutoff	Draws per Chain Used in Rhat	Between-Chain Variance Component	Estimated Marginal Posterior Variance	Gelman-Rubin Rhat
log_alpha	4	10000	10000	1.85254	.000508972	Rhat within cutoff	Yes	1.0100	10000	5.08972	1.85287	1.0001
log_beta	4	10000	10000	0.61807	.000309563	Rhat within cutoff	Yes	1.0100	10000	3.09563	0.61832	1.0002

Table 5. Classical Gelman-Rubin R-hat Results for the Worked Example

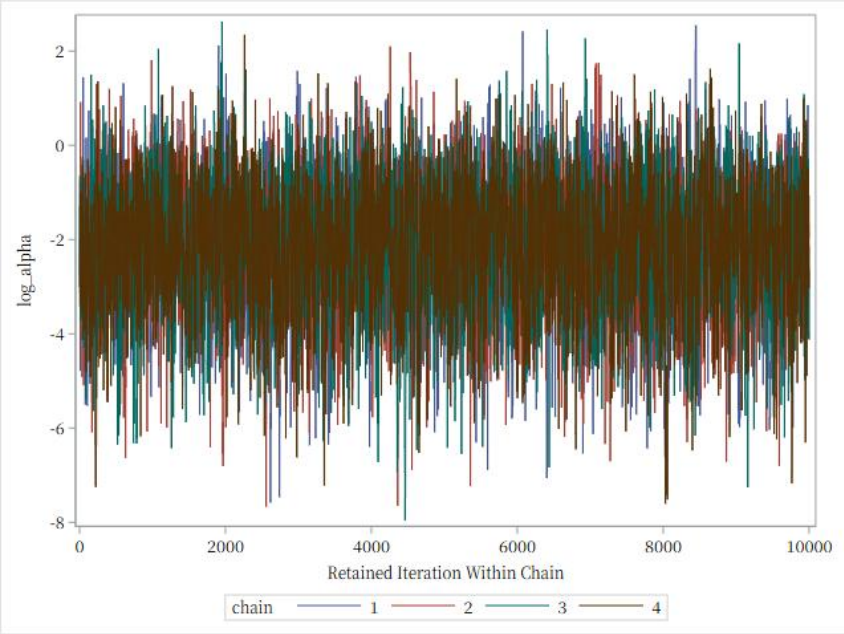

Figure 1. Trace Review Across Independent Chains: log_alpha

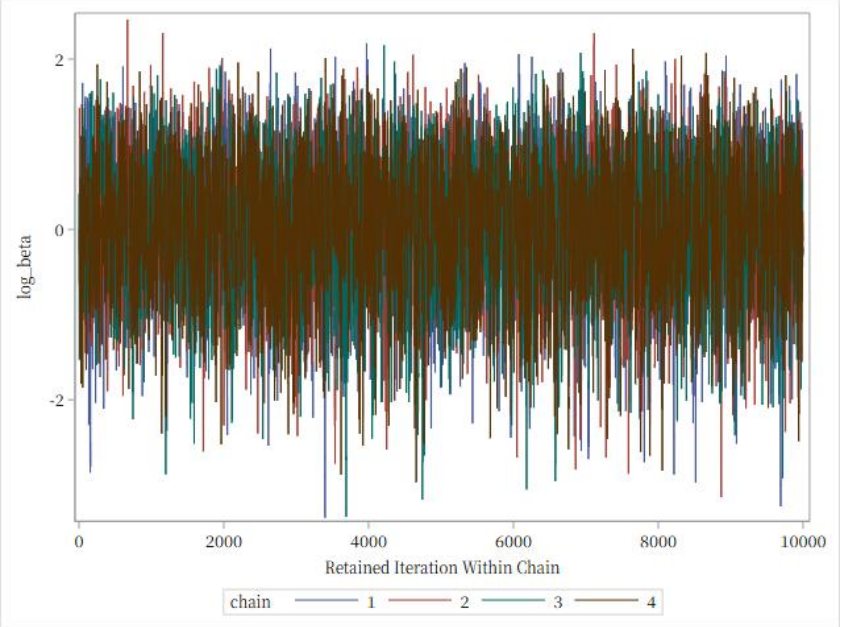

Figure 2. Trace Review Across Independent Chains: log_beta

SIMULATION FRAMEWORK

The same BLRM engine is also used for design evaluation. In practice, simulation is the main tool for calibrating a design before trial start because it reveals how the chosen prior, EWOC threshold, target interval, starting dose, cohort size, and stopping rules behave under alternative truth scenarios. For a single-agent BLRM, simulation is therefore part of the design workflow rather than a separate optional exercise.

The macro blrm_sim_one generates one virtual trial under a user-specified truth scenario. At each step, the macro simulates cohort-level DLT outcomes at the current dose, refits the single-chain BLRM to the cumulative data, applies the EWOC-based decision rule, and selects the next dose subject to the dose-skipping convention. The trial stops when the maximum sample size is reached or when no candidate dose remains acceptable under the EWOC criterion.

The macro blrm_sim_n repeats that process over many simulated trials and stores trial-level outputs such as DLT_ALL, EWOC_ALL, and SIM_ALL. These datasets preserve the path of simulated conduct, including patient allocation, observed DLT counts, dose decisions, and trial-level stopping outcomes. The simulation macros deliberately use the prequalified single-chain engine for computational feasibility and do not repeat full within-chain and between-chain diagnostics for every simulated interim fit.

The macro blrm_sim_sum converts those trial-level outputs into design-level operating characteristics. Typical summaries include the probability of selecting the correct dose, the probability of selecting an acceptable dose, the probability of selecting an over-toxic dose, and the average number of patients treated above the overdosing threshold. Before a large operating-characteristic study, the chosen burn-in and retained iterations should be prequalified with the full multi-chain workflow in representative sparse-data and boundary-sensitive scenarios, using both within-chain and between-chain diagnostics.

Display 3 shows representative SAS code for repeated simulation and operating-characteristic summarization. The example below defines a truth scenario, runs repeated simulations with blrm_sim_n, and then summarizes the resulting operating characteristics with blrm_sim_sum.

```
data dummy_tox;
  dose=1;  true_tox=0.10; output;
  dose=5;  true_tox=0.15; output;
  dose=10; true_tox=0.24; output;
  dose=12; true_tox=0.32; output;
  dose=15; true_tox=0.63; output;
run;

%blrm_sim_n(
  dose_list=%str(1,5,10,12,15),
  ref_dose=10,
  true_tox=dummy_tox,
  start_dose=10,
  max_sample=12, cohortn=3, skip=0,
  mu1=-2.5, mu2=0, v1=4, v2=1, rho=0,
  alpha=0.05, nbi=500, nmc=5000,
  ud=0.16, od=0.33, ewoc=0.30,
  seed=4345, sim_time=1000, debug=0, time_sum=0
);

%blrm_sim_sum(
  sim_data=sim_all,
  dlt_data=dlt_all,
  true_tox=dummy_tox,
  target_range_low=0.16,
  target_range_high=0.33,
  acceptable_low=0.05,
  out_summary=performance_stats,
```

```
    debug=0
);
```

Display 3. Representative SAS Code for Repeated Simulation and Simulation Summarization

Metric	Value	Dose Category Definitions
Prop_Correct_Dose	0.490	Acceptable: [0.05,0.33) | Target: [0.16,0.33) | Over: [0.33, 1]
Prop_Acceptable_Dose	0.880	Acceptable: [0.05,0.33) | Target: [0.16,0.33) | Over: [0.33, 1]
Prop_Over_Toxic_Dose	0.090	Acceptable: [0.05,0.33) | Target: [0.16,0.33) | Over: [0.33, 1]
Avg_Patients_Over_Toxic	1.230	Acceptable: [0.05,0.33) | Target: [0.16,0.33) | Over: [0.33, 1]

Table 6. Illustrative Operating-Characteristic Summary Under the Example Truth Scenario

In this workflow, blrm_sim_n generates trial-level records across simulation batches, and blrm_sim_sum converts those records into design-level summaries. This separation mirrors the interim workflow: one macro performs the computational engine, and another macro turns the results into concise reporting outputs. The simulation settings should be documented together with their prior multi-chain diagnostic qualification.

EXTERNAL IMPLEMENTATION CHECK AGAINST R/JAGS

A matched external implementation check was conducted against an independently coded R/JAGS analysis. The comparison used the same observed data (one DLT among six subjects at 2 mg and no DLT among three subjects at 4 mg), candidate dose grid (2, 4, 6, 8, and 12 mg), reference dose (12 mg), prior mean vector (-0.9, 0), prior covariance matrix diag(4, 1), underdosing and overdosing boundaries (0.15 and 0.35), and EWOC bound (0.25). In the R/JAGS code, theta[1] corresponds to log_alpha and theta[2] corresponds to log_beta in the SAS implementation. Thus, both implementations use the same linear predictor, logit{pi(d)} = log_alpha + exp(log_beta) log(d / 12), and the same bivariate normal prior on (log_alpha, log_beta).

Both implementations used eight independently initialized chains and retained 50,000 posterior draws per chain. The R/JAGS analysis used a 2,000-iteration adaptation period followed by 20,000 burn-in iterations. The SAS analysis specified 20,000 burn-in iterations and 50,000 retained iterations per chain. Since JAGS and PROC MCMC use different simulation engines and tuning mechanisms, their random draws and diagnostic estimates are not expected to agree iteration by iteration. The comparison therefore focuses on posterior region probabilities and the resulting EWOC classifications, while convergence diagnostics are reviewed within each implementation separately.

Table 6 compares posterior probabilities of underdosing, target toxicity, and overdosing from the two implementations.

Dose (mg)	SAS Pr(UD)	SAS Pr(TT)	SAS Pr(OD)	R/JAGS Pr(UD)	R/JAGS Pr(TT)	R/JAGS Pr(OD)
2	0.78765	0.19647	0.01589	0.78964	0.19435	0.01602
4	0.58456	0.35146	0.06399	0.58973	0.34419	0.06608
6	0.45116	0.40091	0.14793	0.45302	0.39857	0.14842
8	0.37088	0.40124	0.22788	0.37255	0.39969	0.22777
12	0.28470	0.36810	0.34721	0.28430	0.36750	0.34820

Table note. UD = underdosing, TT = target toxicity, and OD = overdosing. The region definitions are Pr(pi(d) < 0.15), Pr(0.15 <= pi(d) < 0.35), and Pr(pi(d) >= 0.35), respectively. EWOC classifications and absolute differences are summarized in the text rather than added as further columns.

Table 7. Comparison of Posterior Region Probabilities from the SAS and R/JAGS Implementations

CONCLUSION

This paper presents a single-agent BLRM implementation in SAS that integrates posterior updating, within-chain and between-chain MCMC diagnostics for formal inference, posterior toxicity summarization, EWOC-based decision support, a matched external comparison with an independently coded R/JAGS implementation, and repeated simulation for design evaluation.The main contribution is not a novel statistical design, but rather a practical and reusable programming workflow for dose escalation in phase I oncology trials.

Given the limited public availability of SAS-focused BLRM implementation reports, the workflow presented herein may help address a practical gap for SAS-oriented programming teams seeking a transparent, reusable, and publication-ready foundation.

By organizing the implementation around modular macros, chain-labeled posterior draws, explicit within-chain diagnostic datasets, and a between-chain R-hat output, the framework facilitates transparent review and straightforward downstream reporting. The worked example and simulation workflow demonstrate how BLRM can be operationalized in SAS while distinguishing a toxicity-based model recommendation from the broader cross-functional RP2D decision.

REFERENCES

Gelman, A., and D. B. Rubin. 1992. “Inference from Iterative Simulation Using Multiple Sequences.” Statistical Science, 7(4):457–472.

Geweke, J. 1992. “Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments.” In Bayesian Statistics 4, edited by J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, 169–193. Oxford: Clarendon Press.

SAS Institute Inc. 2020. SAS/STAT User’s Guide: The MCMC Procedure. Cary, NC: SAS Institute Inc.

Guo, X., S. Kent, A. Maity, and W. Zhong. 2025. “Optimization of EWOC Principle in BLRM Design for Phase 1 Oncology Trials.” Journal of Biopharmaceutical Statistics, 35(3):407–423.

Neuenschwander, B., M. Branson, and T. Gsponer. 2008. “Critical Aspects of the Bayesian Approach to Phase I Cancer Trials.” Statistics in Medicine, 27(13):2420–2439.

Neuenschwander, B., S. Roychoudhury, and H. Schmidli. 2016. “On the Use of Co-Data in Clinical Trials.” Statistics in Biopharmaceutical Research, 8(3):345–354.

Widmer, L. A., A. Bean, D. Ohlssen, and S. Weber. 2023. “Principled Drug-Drug Interaction Terms for Bayesian Logistic Regression Models of Drug Safety in Oncology Phase I Combination Trials.” arXiv.

Bhattacharyya, R. 2018. “Innovative Approach for Early Phase Clinical Trials – A Case Study.” PSI 2018 Conference Poster.

Novartis. 2023. “SAP CBLZ945X2101.” Statistical analysis plan available through ClinicalTrials.gov supporting documents.

Weber, S., L. A. Widmer, A. Bean, et al. 2025. OncoBayes2: Bayesian Logistic Regression for Oncology Dose-Escalation Trials. R package manual.

Boehringer Ingelheim. 2025. decider: Decision Making in Multiple-Arm Dose Escalation Trials Using Bayesian Logistic Regression Models. GitHub repository.

Zhang, H., A. Y. Chiang, and J. Wang. 2022. “Improving the Performance of Bayesian Logistic Regression Model with Overdose Control in Oncology Dose-Finding Studies.” Statistics in Medicine, 41(27):5463–5483.

Zhou, H., Y. Yuan, and L. Nie. 2018. “Accuracy, Safety, and Reliability of Novel Phase I Trial Designs.” Clinical Cancer Research, 24(18):4357–4364.

Peng, Zhanwen. 2026. blrm-sas-single-agent: SAS implementation of a single-agent Bayesian logistic regression model for phase I dose escalation. GitHub repository. Available at https://github.com/zhanwenPengbio/blrm-sas-single-agent.

CONTACT INFORMATION

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

Zhanwen Peng

Akeso, Inc.

E-mail: zhanwen.peng@akesobio.com

GitHub: https://github.com/zhanwenPengbio/blrm-sas-single-agent