Journal of Statistical Software

June 2014, Volume 58, Issue 8. *http://www.jstatsoft.org/*

%HPGLIMMIX: A High-Performance SAS Macro for GLMM Estimation

Liang Xie
Microsoft Corp.

Laurence V. Madden
Ohio State University

Abstract

Generalized linear mixed models (GLMMs) comprise a class of widely used statistical tools for data analysis with fixed and random effects when the response variable has a conditional distribution in the exponential family. GLMM analysis also has a close relationship with actuarial credibility theory. While readily available programs such as the GLIMMIX procedure in SAS and the **lme4** package in R are powerful tools for using this class of models, these progarms are not able to handle models with thousands of levels of fixed and random effects. By using sparse-matrix and other high performance techniques, procedures such as HPMIXED in SAS can easily fit models with thousands of factor levels, but only for normally distributed response variables. In this paper, we present the %HPGLIMMIX SAS macro that fits GLMMs with large number of sparsely populated design matrices using the doubly-iterative linearization (pseudo-likelihood) method, in which the sparse-matrix-based HPMIXED is used for the inner iterations with the pseudo-variable constructed from the inverse-link function and the chosen model. Although the macro does not have the full functionality of the GLIMMIX procedure, time and memory savings can be large with the new macro. In applications in which design matrices contain many zeros and there are hundreds or thousands of factor levels, models can be fitted without exhausting computer memory, and 90% or better reduction in running time can be observed. Examples with a Poisson, binomial, and gamma conditional distribution are presented to demonstrate the usage and efficiency of this macro.

Keywords: GLMM, REPL, pseudo-likelihood, SAS.

1. Introduction

Mixed models comprise a class of important statistical tools to estimate variance and covariance parameters, account for repeated measurements and other features of experimental designs, and adjust for over-dispersed data (Stroup 2012). Mixed models extend the classic fixed effect models by including random effects and best linear unbiased predictors for

subjects. The random effect represents a random sample from a hypothetical distribution, and serves as a mechanism to link observations with the same level of random effect via a covariance matrix, so that information from similar observations can be utilized in estimation. Mixed modeling also has a close relationship with actuarial credibility theory. The generalized linear mixed model (GLMM) has attracted considerable attentions during the past two decades, because it extends the linear mixed model to a general framework that accommodates a rich set of distributions from the exponential family, so that non-normally distributed data such as counts and binary observations can be modeled appropriately. Readily available commercial or free software packages, such as the GLIMMIX procedure from SAS Institute Inc. (2011a) and the **lme4** package (Bates, Maechler, Bolker, and Walker 2014) in R (R Core Team 2014) make GLMMs increasingly popular to the research community. GLMMs have been widely applied in areas such as biology (Vergara, Aguirre I, and Fernandez-Cruz 2007), ecology (Milsom, Langton, Parkin, Peel, Bishop, Hart, and Moore 2000), small area estimation (Maiti 2001), genetic research (Kerr, Martin, and Churchill 2000), and actuarial science (Antonio and Beirlant 2007; Frees, Young, and Lou 1999; Kaas, Dannenburg, and Goovaerts 1997), to name a few. In many of these applications, however, model fitting is a challenging task because the fixed and random effects may have a large number of levels. This is especially true with molecular biology studies as indicated by Wolfinger *et al.* (2001). Here, we present a macro in SAS for fitting GLMMs to data with large numbers of fixed and random effect levels using sparse-matrix techniques, and compare results with the output of the GLIMMIX procedure in SAS (which does not use sparse-matrix or other high performance techniques).

To understand the computational challenge, it is necessary to review the estimation techniques, which fall into either of the two categories:

1. Linearization of the model based on a Taylor series, such as described in Breslow and Clayton (1993), Wolfinger and O'Connell (1993), Schall (1991).

2. Integral approximation of the GLMM log-likelihood function, such as described in Wolfinger (1993), Pinheiro and Bates (1995), Raudenbush, Yang, and Yosef (2000), Pinheiro and Chao (2006).

The linearization method is more general than the integral-approximation method (in terms of the diversity of models that can be fitted), but may produce more biased variance-covariance and other parameter estimates than found with integral-approximation methods. Stroup (2012) shows, however, that the bias problem is usually of concern only under extreme situations, such as when the number of Bernoulli trials per sampling unit is very small, especially if the number of subjects is small. The linearization method is the focus of this paper because of its generality and how this approach can be incorporated into a high performance computational algorithm.

Schall (1991) and Breslow and Clayton (1993) proposed a method based on the first-order Taylor-series expansion of the inverse link function around the current estimate of fixed and random effects, which is known as a quasi-likelihood based method. It is also known as a so-called penalized quasi-likelihood (PQL) method. Wolfinger and O'Connell (1993) expanded the Taylor-series approach by incorporating a probabilistic approximation based on the Gaussian distribution. This results in a so-called pseudo-likelihood approach because the marginal log-likelihood for the approximating function mimics the structure of a Gaussian log-likelihood. Within this structure, iterative mixed-model estimation is achieved using

likelihood- or restricted-likelihood based methods and iteratively-reweighted-least-squares, essentially coupling and generalizing linear mixed model (LMM) and generalized linear model (GLM) algorithms (Schabenberger and Pierce 2002). This section basically follows Wolfinger and O'Connell (1993) and Stroup (2012).

A GLMM can be expressed as:

$$\mathsf{E}(\mathbf{y}|\gamma) = h(\mathbf{X}\beta + \mathbf{Z}\gamma) = h(\eta) = \mu|\gamma$$

and

$$\mathsf{VAR}(\mathbf{y}|\gamma) = \mathbf{A}^{1/2}\mathbf{R}\mathbf{A}^{1/2},$$

where $\mathbf{y}$ is the response vector, $\mathbf{X}$ is the fixed effects design matrix, $\mathbf{Z}$ is the random effects design matrix, β is the vector of fixed effects parameters, γ is the vector of random effects, μ is the vector of expected values, η is the vector of linear predictors conditional on the random effects, and $h()$ is the inverse link function $g^{-1}()$. It is assumed that $\gamma \sim N(0, \mathbf{G})$, where $\mathbf{G}$ is the variance-covariance matrix of the random effects. We are mostly concerned about variance-component models, which correspond to a diagonal $\mathbf{G}$ matrix, but the approach is applicable to a wider class of models. The variance (or variance-covariance) of $\mathbf{y}$ conditional on the random effects is defined through two matrices. $\mathbf{A}$ is a diagonal matrix whose elements represent the variance function for $h(\eta)$ (dependent on the assumed conditional distribution, and calculated at μ), and $\mathbf{R}$ is a scaling matrix.

In the nominal situation, $\mathbf{R}$ is a diagonal matrix with elements ϕ, a "residual-type" scaling term; for some conditional distributions in the exponential family, such as the binomial and Poisson, $\phi \equiv 1$. For other conditional distributions (e.g., gamma, normal, negative binomial), ϕ is unknown and must be estimated. Over-dispersion with the binomial and the Poisson distribution can be accounted for by allowing ϕ to be an unknown parameter that is estimated; this is equivalent to holding ϕ fixed at the theoretical value for the conditional distribution and multiplying it by an over-dispersion parameter. In this over-dispersion situation with the binomial and Poisson distribution, the estimation becomes a quasi-likelihood method, because the "likelihood" no longer corresponds to a known distribution (Stroup 2012). $\mathbf{R}$ can also be generalized to a non-diagonal matrix as one approach to account for correlations of the observations within subjects.

Define the first order derivative of the inverse link function $h()$ evaluated at a given estimate of linear predictor effects β, γ as:

$$\left(\frac{\partial h(\eta)}{\eta}\right)_{\eta=\hat{\eta}} = h^{'}(\hat{\eta})$$

and

$$\hat{\mathbf{D}} = \mathrm{diag}\left[h^{'}(\hat{\eta})\right].$$

Then the first-order Taylor-series expansion of the GLMM at a given estimate of linear predictor effects is:

$$h(\eta) \cong h(\hat{\eta}) + \hat{\mathbf{D}}(\eta - \hat{\eta}).$$

Here, the hats refer to the current estimate of the parameter (or parameter vector) in an iterative process. Rearranging terms, we have

$$\mathbf{X}\beta + \mathbf{Z}\gamma = \mathbf{X}\hat{\beta} + \mathbf{Z}\hat{\gamma} + \hat{\mathbf{D}}^{-1}\left[h(\eta) - h(\hat{\eta})\right]$$

Following the idea from the iterative re-weighted least squares algorithm with a GLM, the pseudo-variable is defined as

$$\mathbf{y}^* = \hat{\eta} + \hat{\mathbf{D}}^{-1}[\mathbf{y} - h(\hat{\eta})], \tag{1}$$

where $\hat{\eta} = \mathbf{X}\hat{\beta} + \mathbf{Z}\hat{\gamma}$. It follows that the conditional expected value and variance are given by Stroup (2012):

$$\mathsf{E}(\mathbf{y}^*|\gamma) = \mathbf{X}\hat{\beta} + \mathbf{Z}\hat{\gamma} + \hat{\mathbf{D}}^{-1}[h(\eta) - h(\hat{\eta})]$$

and

$$\mathsf{VAR}(\mathbf{y}^*|\gamma) = \hat{\mathbf{D}}^{-1}\mathbf{A}^{1/2}\mathbf{R}\mathbf{A}^{1/2}\hat{\mathbf{D}}^{-1}.$$

With the pseudo-likelihood approach, it is assumed that $\mathbf{y}^* \mid \gamma$ has a normal distribution. Using $\mathbf{y}^*$ as the response variable, pseudo-likelihood estimation of a GLMM is achieved within the framework of a linear mixed model, with weights defined as $\hat{\mathbf{W}}$, a diagonal matrix with elements as $\mathbf{A}^{-1}\hat{\mathbf{D}}^2$. Under the canonical link function, $\mathbf{W} = \mathbf{A}^{-1}$. Estimates of β and predictions of γ are obtained by solving the GLMM equations:

$$\mathbf{H}\begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} \mathbf{X}\hat{\mathbf{S}}^{-1}\mathbf{y}^* \\ \mathbf{Z}\hat{\mathbf{S}}^{-1}\mathbf{y}^* \end{pmatrix},$$

where

$$\mathbf{H} = \begin{pmatrix} \mathbf{X}^\top\hat{\mathbf{S}}^{-1}\mathbf{X} & \mathbf{X}^\top\hat{\mathbf{S}}^{-1}\mathbf{Z} \\ \mathbf{Z}^\top\hat{\mathbf{S}}^{-1}\mathbf{X} & \mathbf{Z}^\top\hat{\mathbf{S}}^{-1}\mathbf{Z} + \hat{\mathbf{G}}^{-1} \end{pmatrix}$$

and $\hat{\mathbf{S}} = \hat{\mathbf{D}}^{-1}\hat{\mathbf{A}}^{1/2}\hat{R}\hat{\mathbf{A}}^{1/2}\hat{\mathbf{D}}^{-1} = \hat{\mathbf{W}}^{-1/2}\hat{\mathbf{R}}\hat{\mathbf{W}}^{-1/2}$.

Note that the "marginal" variance of $\mathbf{y}^*$ is

$$\begin{aligned} \mathsf{VAR}(\mathbf{y}^*) &= \mathbf{V} \\ &= \mathbf{Z}\mathbf{G}\mathbf{Z}^\top + \hat{\mathbf{D}}^{-1}\mathbf{A}^{1/2}\mathbf{R}\mathbf{A}^{1/2}\hat{\mathbf{D}}^{-1} \\ &= \mathbf{Z}\mathbf{G}\mathbf{Z}^\top + \mathbf{W}^{-1/2}\mathbf{R}\mathbf{W}^{-1/2}. \end{aligned}$$

Given the probability approximation as above, the objective function is the Gaussian log-likelihood function for the pseudo-variable $\mathbf{y}^*$:

$$pl = -\frac{1}{2}\log \mid \mathbf{V} \mid -\frac{1}{2}(\mathbf{y}^* - \mathbf{X}\beta)^\top\mathbf{V}^{-1}(\mathbf{y}^* - \mathbf{X}\beta) - \frac{n}{2}\log(2\pi) \tag{2}$$

and the restricted pseudo-log-likelihood function is:

$$pl_R = -\frac{1}{2}\log \mid \mathbf{V} \mid -\frac{1}{2}\log\left(\mid \mathbf{X}^\top\mathbf{V}^{-1}\mathbf{X} \mid\right) - \frac{1}{2}\left(\mathbf{r}^\top\mathbf{V}^{-1}\mathbf{r}\right) - \frac{n-p}{2}\log(2\pi) \tag{3}$$

where $\mathbf{r} = (\mathbf{I} - \mathbf{X}(\mathbf{X}^\top\mathbf{V}^{-1}\mathbf{X})\mathbf{X}^\top\mathbf{V}^{-1})\mathbf{y}^*$, n denotes the sum of the frequencies used in the analysis and p is the rank of $\mathbf{X}$.

In Wolfinger and O'Connell (1993), the uses of Equations 2 and 3 (with the GLMM equations) were originally called the PL and REPL algorithms, respectively. In the `GLIMMIX` procedure of SAS, the PL algorithm is referred to as MSPL (maximum subject-specific pseudo-likelihood), while the REPL algorithm is called RSPL (restricted subject-specific pseudo-likelihood). Note that the elements of $\mathbf{R}$ can either be held constant based on the conditional distribution (consistent with Breslow and Clayton 1993), or be estimated (consistent with Wolfinger and O'Connell 1993). The `GLIMMIX` procedure uses the MSPL and RSPL labels for the linearization

estimation methods whether or not the $\mathbf{R}$ matrix is estimated. This is discussed further below. A general label for all of these approaches in this paragraph is *linearization*.

Estimation of a linearized GLMM follows a doubly iterative algorithm, as indicated by SAS Institute Inc. (2011a). In a doubly iterative algorithm, a simpler model, a LMM, is derived from the original more complex GLMM; here the pseudo-variable ($\mathbf{y}^*$) is calculated and is fitted to data as a LMM using the above-described pseudo log-likelihood and mixed-model equations. For most LMMs, this is an iterative process, known as the inner iteration. Using the parameter estimates and predictions of the random effects obtained after convergence, $\mathbf{y}^*$ is re-calculated (the outer iteration) and a LMM is again fitted to the data. The outer iterations continue (with the corresponding inner iterations at each step) until a preset convergence criterion is met or the maximum number of iterations is attained. The deviance based on the assumed conditional distribution is then calculated. The algorithm is outlined in Section 2.

When either the fixed effect design matrix $\mathbf{X}$ or the random effect design matrix $\mathbf{Z}$ has many columns, solving the mixed model equations will be extremely time consuming and memory intensive, especially when determining the generalized inverse of the matrices, see SAS Institute Inc. (2011a). The time complexity will roughly be about $O(k^3)$ and the space complexity will roughly be about $O(k^2)$, where k is the rank of the matrix $\mathbf{H}$. What makes the new macro outperform the `GLIMMIX` procedure in terms of speed and memory consumption is the use of sparse-matrix techniques and special optimization methods in the inner iterations of the doubly iterative algorithm. This is accomplished by calling the new `HPMIXED` procedure for the inner-iteration calculations instead of using a more traditional LMM algorithm not adapted for large scale problems. `HPMIXED` is specifically developed for linear mixed models with large numbers of sparsely populated columns in the $\mathbf{X}$ and $\mathbf{Z}$ matrices.

In a mixed model with fixed and/or random effects that have large number of levels, the resulting mixed model equation matrices are very large, but often extremely sparse in the sense that most of the elements are 0. For a typical variance-component mixed model with many factor levels, close to 99% of the elements may be 0. Sparse-matrix techniques exploit this fact by representing a matrix not as a complete two dimensional array, but as a set of nonzero elements and their location (row and column) within the matrix. The `HPMIXED` procedure in SAS, in particular, employs the compressed sparse row (CSR) representation of a sparse matrix, where nonzero elements are stored row by row in `(value, col_ind, row_ptr)` format where `value` is an array of the (left-to-right, then top-to-bottom) non-zero values of the matrix; `col_ind` is the column index corresponding to the values; and `row_ptr` is the list of value indexes where each row starts. CSR is efficient for row-wise arithmetic operations which is exactly how the likelihood calculations for mixed model are conducted.

Several optimization methods are possible for the linear mixed model fit, and the default in `HPMIXED` is dual quasi-Newton, which only requires first derivatives of the (restricted) pseudo-log-likelihood. `HPMIXED` also provides several optional optimization techniques to choose from when solving the pseudo-log-likelihood function, some of which require the calculation of the second derivatives. As the default for the `%HPGLIMMIX` macro, the `HPMIXED` default is replaced with the Newton-Raphson with ridging optimization method. Table 1 shows available optimization techniques and whether second-order derivatives are required. These are chosen with the `TECH=` option in the macro (see last example for a demonstration). However, `HPMIXED` does not actually calculate the true second derivative (or the observed information matrix). Instead, the so-called average information matrix is calculated, which is much less computationally demanding, and can be more stable (Gilmour, Thompson, and Cullis 1995).

Algorithm	Full name	Gradient	Hessian
LEVMAR	Levenberg-Marquardt	Yes	Yes
TRUREG	Trust Region	Yes	Yes
NEWRAP	Newton-Raphson with Line Search	Yes	Yes
NRRIDG	Newton-Raphson Ridge	Yes	Yes
QUANEW	Quasi-Newton	Yes	No
DBLDOG	Double-Dogleg	Yes	No
CONGRA	Conjugate Gradient	Yes	No
NMSIMP	Nelder-Mead Simplex	No	No

Table 1: List of optimization algorithms and derivatives required.

2. Outline of linearization algorithm for GLMMs

The specific algorithm implemented in this macro as well as the description below follows that of Wolfinger and O'Connell (1993).

1. Set up variance function and deviance function based on Table 2 according to specified conditional distribution.
2. Use the original data as initial estimate of μ, $\hat{\mu}$. Adjustment (correction factor), as in Table 3, may be applied to y in order to apply the link function (e.g., avoid the log of 0). Correction factor is set to 0.5 by default.
3. Compute pseudo-variable $\mathbf{y}^*$ according to Equation 1.
4. Diagonal weight matrix $\hat{\mathbf{W}} = \hat{\mathbf{A}}^{-1}\hat{\mathbf{D}}^2$ is also computed based on $\hat{\mu}$ and specified distribution.
5. In the inner iterations, use REML to estimate components of covariance matrices $\mathbf{G}, \mathbf{R}$ (or just $\mathbf{G}$, if there is not a free scale parameter with the specified distribution and one does not wish to adjust for over-dispersion after random effects are in the model) and solve the mixed model equation for fixed and random effects.
6. Obtain the maximum difference of estimates for covariance parameters and fixed effect parameters between the current and previous outer iteration. Convert the difference to a relative scale by dividing by the magnitude of the corresponding parameter estimate.
7. If the max (relative) difference is larger than a threshold (e.g., `1E-8`), update μ using the inverse link function with newly estimated fixed and random effects. Then go back to Step 3.
8. If the max (relative) difference is smaller than the threshold, claim convergence, calculate deviance, and assemble requested statistics.

3. %HPGLIMMIX macro program

%HPGLIMMIX largely follows the structure of the now obsolete %GLIMMIX macro SAS Institute Inc. (2007), from which it is derived, and has almost the same set of input parameters.

Distribution	Variance function	Deviance
Normal	1	$\sum_i w_i(y_i - \mu_i)^2$
Binary	$\mu(1-\mu)$	$2\sum_i w_i \left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i)\right]$
Binomial	$\mu(1-\mu)$	$2\sum_i w_i \left[y_i \log(y_i/\mu_i) + (1-y_i)\log\left(\frac{y_i-\mu_i}{\mu_i}\right)\right]$
Poisson	μ	$2\sum_i \left[y_i \log(y_i/\mu_i)\right]$
Gamma	μ^2	$-2\sum_i w_i \left[\log(y_i/\mu_i) - (y_i-\mu_i)/\mu_i\right]$
Inv. Gaussian	μ^3	$\sum_i w_i \left[(y_i-\mu_i)^2/(y_i\mu_i^2)\right]$
Geometric	$\mu + \mu^2$	$2\sum_i w_i \left[y_i \log(y_i/\mu_i) - (y_i+w_i)\log\left(\frac{y_i+w_i}{\mu_i+w_i}\right)\right]$

Table 2: List of variance and deviance functions.

Distribution	Apply correction factor (CF)
Binomial or binary	(Response+CF)/(1+2 · CF)
Binomial using event/trail	(Event+CF)/(Trail + 2 · CF)
All others	Response + CF

Table 3: Correction factors.

Some parameters are dropped that do not apply to the `HPMIXED` procedure (see below), and some are added (such as one for the optimization method in the inner iterations (`TECH=`). All of the above listed items in Section 2 are automatically carried out by the macro.

The `%GLIMMIX` macro does pseudo-likelihood estimation or restricted pseudo-likelihood estimation of GLMMs, which operates by repeatedly calling the `MIXED` procedure with the pseudo-variable and weights updated with each call to the `MIXED` procedure. Later, SAS Institute Inc. put the functionality of this macro, together with many other features, into the `GLIMMIX` procedure. However, sparse-matrix techniques are not incorporated into the `GLIMMIX` procedure. Thus, we used `%GLIMMIX` as a template for the development of a new macro that repeatedly calls `HPMIXED` procedure instead of calling the `MIXED` procedure. In addition, many segments of the data processing code in the macro have been rewritten to achieve the maximum efficiency in data processing and updating when using big data. `HPMIXED` only supports REML estimation; thus, the new macro can only perform restricted pseudo-likelihood methods (RSPL) to fit GLMMs using the linearization approach. Additionally, the computational part of the new macro is significantly modified to both accommodate the syntax difference between `HPMIXED` and `MIXED` procedures and improve efficiency, especially for larger data sets with many observations. The `HPMIXED` procedure has only a subset of the options available in the more general `MIXED` procedure, and default settings are different in some cases (see below). It is assumed that the user has general familiarity with the syntax of the `GLIMMIX` procedure for fititing GLMMs and the `MIXED` or `HPMIXED` procedures of SAS for fitting linear mixed models.

Users can invoke the macro by calling `%HPGLIMMIX`. The list below explains key parameters in

Distribution	Abbreviation	Default link
Binary	bi	Logit
Binomial	b	Logit
Normal	n	Identity
Poisson	p	Log
Gamma	g	Reciprocal
Invgaussian	ig	Power(−2)
Geometric	ge	Log
User	u	User-specified

Table 4: Supported distributions and default link functions.

the syntax that will be used most often, while explanation of the full list of parameters is in the `.sas` program file and basically follows the instructions for the `%GLIMMIX` macro in SAS Sample 25030.

1. `DATA=` specifies the data set you are using. It can either be a regular input data set or the `_DS` data set from a previous call to `%HPGLIMMIX`. The latter is used to specify starting values for `%HPGLIMMIX` and should be accompanied by the `INITIAL` keyword option in the `OPTIONS=` option (see below for description of `OPTIONS`).

2. `STMTS=` specifies `HPMIXED` procedure statements for the analysis, separated by semicolons and listed as a single argument to the `%str()` macro function. Statements may include any of the following: `CLASS, MODEL, RANDOM, REPEATED, PARMS, ID, TEST`. Syntax and options for each statement are exactly as in the `HPMIXED` procedure documentation. Most aspects of the GLMM specification (in terms of fixed and random effects, continuous versus categorical [dummy] variables, and over-dispersion) are given with these statements. Unlike with the `GLIMMIX` procedure, the link function and conditional distribution are not given in the `MODEL` statement but are specified with separate options. The `TEST` statement is explained below.

3. `ERROR=` specifies the distribution of y conditional on the random effects (sometimes known as the error distribution). When you specify `ERROR=USER`, you must also provide the `ERRVAR=` and `ERRDEV=` options. The default conditional distribution is binomial. Valid types and their abbreviations are listed in Table 4.

4. `LINK=` specifies the link function. Valid types are `logit, probit, cloglog, loglog, identity, power(), log, exp, reciprocal, nlin, and user`. The default link function for each error distribution is listed in Table 4. The user should see the `.sas` program for more details.

5. `OPTIONS=` specifies `%HPGLIMMIX` macro options separated by spaces. For example, key word `INITIAL` specifies that the input data set is actually the `_DS` data set from a previous call to `%HPGLIMMIX`. This allows you to restart a problem that stopped or to specify starting values. For a full list of available keywords, refer to the `.sas` program.

6. `PROCOPT=` specifies the options used by the `HPMIXED` procedure statement. Refer to the `HPMIXED` procedure documentation for more information.

7. TECH= specifies the optimization algorithm for covariance component estimation, default is NRRIDG (Newton-Raphson ridge). Available algorithms are listed in Table 1.

There are some important differences between the %HPGLIMMIX macro and the GLIMMIX procedure, even if the same linearization (pseudo-likelihood) algorithm is used for both procedures, mostly due to the difference between the HPMIXED and the GLIMMIX procedures. Because of differences between the HPMIXED and the MIXED procedure, there are also a few differences between the %GLIMMIX and %HPGLIMMIX macros.

First, the syntax between HPMIXED and GLIMMIX procedures has some differences. For example, the statements COVTEST, LSMESTIMATES, SLICE and FREQ in GLIMMIX are not supported in HPMIXED (making them, therefore, unavailable in the %HPGLIMMIX macro), while the LSMEANS and CONTRAST statements in HPMIXED do not provide the same level of functionality as those in GLIMMIX. On the other hand, HPMIXED does not automatically produce global tests of fixed effects (main effects or interactions) in order to reduce computational time and memory usage for big data problems; for situations with huge numbers of factor levels, overall F tests are often not of value. F tests of main effects and interactions, when desired, are specified with TEST statements in HPMIXED and in the %HPGLIMMIX macro; these tests are automatically obtained with GLIMMIX procedure. An example of the TEST statement is given in Section 4.3.

Second, %HPGLIMMIX supports the REPEATED statement in HPMIXED procedure for modeling the so-called R-side (residual) variation; in contrast, the GLIMMIX procedure uses the RANDOM _RESIDUAL_ statement for the same or similar purpose (depending on the type of GLMM). However, there are some important differences that must be kept in mind between the macro and the procedure in this regard, depending on the selected conditional distribution; %HPGLIMMIX follows the same convention as the obsolete %GLIMMIX in this regard. If one fits a model without a free scale parameter, such as the Poisson, binary, or binomial conditional distribution, there is no residual variance term in the GLIMMIX procedure (because the conditional residual variance is fully defined as a function of the mean). In terms of the GLMM, the $\mathbf{R}$ matrix has no unknown parameters, as discussed in the introduction. But with the HPMIXED (or MIXED) procedure, there is always a residual term. So, in essence, there is one more variance (or variance-covariance) parameter with the macro than with the procedure for these conditional distributions. With the macro, one must force the "last" variance term (residual variance) to equal 1 in order to perform a pseudo-likelihood analysis and duplicate the model (and the results) of the GLIMMIX procedure (for those conditional distributions without a free scale parameter). This difference is demonstrated in Sections 4.1 and 4.2. In Section 4.1, it is shown that if an extra scale parameter is desired with the models to deal with over-dispersion that is not accounted for with (conditional) random effects, the statement: RANDOM _RESIDUAL_ has to be explicitly specified in GLIMMIX procedure. In contrast, with the %HPGLIMMIX macro, this scale parameter is automatically estimated. In Section 4.2, the opposite case is demonstrated, where there are four variance parameters with the %HPGLIMMIX macro but with the 4th variance-covariance parameter held at 1 by specifying HOLD=4 option in the RANDOM statement, and only three explicit variance parameters with the GLIMMIX procedure with the scale parameter automatically hold at 1.

For other conditional distributions which have a free (residual) scale parameter (e.g., gamma, inverse Gaussian), nothing special has to be done with the macro (or with the procedure); that is, the number of variance-covariance parameters match up naturally between the macro and the procedure.

Third, HPMIXED uses the residual denominator degrees of freedom (df) for tests of fixed effects. The only other option is to use an infinite df, which means that t and F tests become z and chi-square tests, respectively. In the GLIMMIX and MIXED procedures, several df calculation or estimation methods are allowed, and the residual method is not the default. Thus, for direct compatibility in denominator df between the new macro and the procedure (or the %GLIMMIX macro), one needs to use the ddfm=residual option in the GLIMMIX procedure (or in the %GLIMMIX macro), as shown in the first example below.

Fourth, it is routine in data analysis for models to be fitted with an over-parameterized fixed effect component ($\mathbf{X}\beta$), which means that there is an infinite number of fixed effect parameters with the same model fit; only certain linear combinations of the parameters are estimable and are unique. This happens typically when classification variables (factors) are in the model. In MIXED, HPMIXED, and GLIMMIX, the generalized inverse used in the mixed-model equations results in one of the factor levels (the reference level) being "estimated" as 0. With MIXED and GLIMMIX procedures, by default, the 0 is obtained for the last factor level, but with HPMIXED, the order of 0 estimates is almost random and cannot be controlled by the user. Thus, for an over-parameterized model, the estimates of β from HPMIXED may differ from those in GLIMMIX or MIXED, although the estimatable functions will be the same (e.g., least squares means, contrasts).

4. Examples

In this section, several examples are used to demonstrate key features of the new high performance macro. First, in Section 4.1 the new macro is shown to be in agreement with the now obsolete %GLIMMIX macro for the same model. In Section 4.2, we show how one needs to fix the residual variance at 1 in the %HPGLIMMIX macro code when fitting a model with a conditional binomial distribution. In comparison, this is automatically determined in the procedure. In the third example, we show that the new macro saves tremendous amount of time when fitting a large-scale GLMM. In this example, a mixed model with a gamma conditional distribution is used where the fixed effect design matrix has 4513 columns and the random effect design matrix has 3054 columns, ending up with mixed-model equations with more than 7500 columns in total. The total running time using the macro is less than 2.5% compared to the GLIMMIX procedure (67 minutes vs. 2714 minutes). Memory consumption using the new macro is also a tiny fraction of the procedure in this case.

4.1. Agreement between estimates from GLIMMIX procedure and the macro

In this example, the ship data from McCullagh and Nelder (1989) are used, which are available online in SAS Knowledge Base sample 25030 from SAS Institute Inc. (2007). For convenience, the data are listed below. Here, we show that the estimation from %HPGLIMMIX is in agreement with the GLIMMIX procedure as well as the SAS Institute Inc. provided %GLIMMIX macro using the restricted pseudo-likelihood algorithm.

```
data work.ship;
   length type $1. year $7. period $8.;
   input type year period service y;
   datalines;
B  1965-69  1975-79   9.9218  53
```

```
C  1965-69  1975-79   6.5162   1
D  1965-69  1975-79   5.2575   0
E  1965-69  1975-79   6.0799   7
A  1965-69  1975-79   6.9985   4
A  1965-69  1960-74   6.9985   3
B  1965-69  1960-74  10.2615  58
C  1965-69  1960-74   6.6606   0
D  1965-69  1960-74   5.6630   0
E  1965-69  1960-74   6.6708   7
A  1970-64  1960-74   7.3212   6
B  1970-64  1960-74   8.8628  12
C  1970-64  1960-74   6.6631   6
D  1970-64  1960-74   5.8551   2
E  1970-64  1960-74   7.0536   5
A  1970-64  1975-79   8.1176  18
B  1970-64  1975-79   9.4803  44
C  1970-64  1975-79   7.5746   2
D  1970-64  1975-79   7.0967  11
E  1970-64  1975-79   7.6783  12
A  1975-69  1975-79   7.7160  11
B  1975-69  1975-79   8.8702  18
C  1975-69  1975-79   5.6131   1
D  1975-69  1975-79   7.6261   4
E  1975-69  1975-79   6.2953   1
A  1960-64  1960-74   4.8442   0
B  1960-64  1960-74  10.7118  39
C  1960-64  1960-74   7.0724   1
D  1960-64  1960-74   5.5255   0
E  1960-64  1960-74   3.8067   0
A  1960-64  1975-79   4.1431   0
B  1960-64  1975-79   9.7513  29
C  1960-64  1975-79   6.3135   1
D  1960-64  1975-79   4.6540   0
run;
```

We call the `%HPGLIMMIX` macro, just like calling `%GLIMMIX` macro, as:

```
proc sort data=work.ship;
      by descending type;
run;
title "Example 1. Ship data from SAS KB sample 25030";
title2 "Using HPGLIMMIX macro";
%hpglimmix(data=work.ship,
   procopt=order=internal,
   stmts=%str(
      class type year period;
      model y = type / solution;
```

```
      random year|period;
      estimate 'E vs. Others' type -1 -1 -1 -1 4/ divisor=4 cl;
   ),
   error=poisson,
   link=log,
   offset=service
)
run;
```

For comparison purpose, we also estimate the same data using the GLIMMIX procedure. Note that the ddfm=residual option was added to the model statement in the procedure to obtain the residual denominator degrees of freedom. Without this option, different df would be obtained with the procedure (the default df method in the procedure depends on the model structure that is selected) and with the %HPGLIMMIX macro, although the model fits would be the same.

```
title2 "Using PROC GLIMMIX";
proc glimmix data=work.ship order=data;
      class type year period;
      model y = type / solution  d=poisson link=log
                       offset=service  ddfm=residual;
      random year|period;
      estimate 'E vs. Others' type 4 -1 -1 -1 -1
                              / divisor=4 cl;
          random _residual_;
run;
title;
title2;
```

The PROCOPT statement can have many purposes for controlling options in the HPMIXED statement called by the macro (see the HPMIXED procedure documentation). For the %HPGLIMMIX macro, the PROCOPT=ORDER=INTERNAL option is used to specify the order in which to sort the levels of the classification variables listed in the CLASS statement. The sorted order of the classification variable levels from the %HPGLIMMIX macro may be different from that from the GLIMMIX procedure depending on which option you choose. With the %GLIMMIX macro, one uses the PROCOPT=ORDER=DATA option (because MIXED has a different default ordering compared to HPMIXED). The ORDER=DATA option is also used with the GLIMMIX procedure, because the GLIMMIX and MIXED procedures use the same convention for ordering factor levels. It should be pointed out that the ordering of factor levels is often of concern only when the investigator needs the individual parameter estimates for the over-parameterized model. Often, only linear combinations of parameters (such as least squares means or contrasts) are required, and these will not be affected by the parameterization and reference level chosen.

The OFFSET option is used for defining an offset variable in the fixed effect linear predictor (a predictor variable with a parameter equal to 1). Note that the %HPGLIMMIX macro only specified one RANDOM statement corresponding to the factors of interests, but GLIMMIX added another RANDOM statement: RANDOM _RESIDUAL_. This is because, as mentioned previously, the HPMIXED procedure automatically estimates a scale parameter, but for a Poisson conditional distribution, the scale parameter is fixed at 1 by default for the GLIMMIX procedure,

and in order to make the GLIMMIX procedure estimate the same statistical model, this second RANDOM statement is required.

Note that the code for the %GLIMMIX macro is given at http://support.sas.com/kb/25/030.html. To obtain the same denominator df as with the new macro, one uses ddfm=residual for the model statement. The results are identical with the results shown below, but are not shown to save space.

The SAS log shows:

```
1342  proc sort data=work.ship;
1343        by descending type;
1344  run;

NOTE: Input data set is already sorted, no sorting done.
NOTE: PROCEDURE SORT used (Total process time):
      real time           0.00 seconds
      user cpu time       0.00 seconds
      system cpu time     0.00 seconds
      memory              116.18k
      OS Memory           15032.00k
      Timestamp           03/17/2013 12:49:10 PM

1345  title "Example 1. Ship data from SAS KB sample 25030";
1346  title2 "Using HPGLIMMIX macro";
1347  %hpglimmix(data=work.ship,
1348     procopt=order=internal,
1349     stmts=%str(
1350        class type year period;
1351        model y = type / solution;
1352        random year|period;
1353        estimate 'E vs. Others' type -1 -1 -1 -1 4/ divisor=4 cl;
1354     ),
1355     error=poisson,
1356     link=log,
1357     offset=service
1358  )

       The HPGLIMMIX Macro

Data Set            : WORK.SHIP
Error Distribution  : POISSON
Link Function       : LOG
Response Variable   : Y

Job Starts at : 17MAR2013:12:49:10
    HPGLIMMIX Iteration History

Iteration    Convergence criterion
```

```
    1            2            <1 sec
    2            0.209017514  <1 sec
    3            0.037780229  <1 sec
    4            0.0012697992 <1 sec
    5            0.0001319758 <1 sec
    6            0.0000395792 <1 sec
    7            0.0000124198 <1 sec
    8            3.9708249E-6 <1 sec
    9            1.3316894E-6 <1 sec
   10            2.0069757E-8 <1 sec
   11            2.474803E-13 <1 sec

Output from final Proc HPMixed run:
Job Ends at : 17MAR2013:12:49:14
1359  run;
1360
1361  title2 "Using PROC GLIMMIX";
1362  proc glimmix data=work.ship order=data;
1363        class type year period;
1364        model y = type / solution  d=poisson link=log
1365                         offset=service  ddfm=residual;
1366        random year|period;
1367        estimate 'E vs. Others' type 4 -1 -1 -1 -1
1368                                / divisor=4 cl;
1369        random _residual_;
1370  run;

NOTE: Convergence criterion (PCONV=1.11022E-8) satisfied.
NOTE: Estimated G matrix is not positive definite.
NOTE: PROCEDURE GLIMMIX used (Total process time):
      real time           0.14 seconds
      user cpu time       0.01 seconds
      system cpu time     0.04 seconds
      memory              1562.10k
      OS Memory           15032.00k
      Timestamp           03/17/2013 12:49:14 PM

1371  title;
1372  title2;
```

For this small data set, %HPGLIMMIX takes more outer iterations to converge compared to the GLIMMIX procedure and compared to the %GLIMMIX macro from SAS (latter output or log not shown here). This may simply reflect different default starting values for HPMIXED, GLIMMIX and MIXED. GLIMMIX and MIXED use the MIVQUE0 algorithm for starting values for random effects, and GLIMMIX uses the GLM solution for the starting values of the fixed effect parameters; HPMIXED uses the EM-REML algorithm instead for starting values, see SAS Institute Inc. (2011a). Also, the sparse-matrix methods may not be efficient for small data

sets with small numbers of fixed or random effects. That is, the increased computational load of producing the sparse-matrix formulation of the matrices may not be offset until the number of levels of fixed or random effects reaches a certain minimum value (depending on the sparseness of the matrices), relative to the calculations made directly with the original matrices. Examining the results below, we are assured that the estimates of parameters are identical as reported by SAS on-line for the now obsolete `%GLIMMIX` macro, as well as from `GLIMMIX` procedure.

The section below shows parameter estimates output from `%HPGLIMMIX` macro:

Parameter Search

CovP1	CovP2	CovP3	CovP4	Objective Function
0.1174	0.07066	1.11E-10	1.6702	82.307618549

Covariance Parameter Estimates

Cov Parm	Estimate
year	0.1174
period	0.07066
year*period	0
Residual	1.6702

Solution for Fixed Effects

Effect	Type	Estimate	Standard Error	DF	tValue	Pr>|t|	Alpha	Lower	Upper
Intercept		-5.6799	0.3286	29	-17.28	<.0001	0.05	-6.3519	-5.0078
type	A	0	.	.	.	.	.	.	.
type	B	-0.5798	0.2277	29	-2.55	0.0164	0.05	-1.0454	-0.1141
type	C	-0.6984	0.4248	29	-1.64	0.1110	0.05	-1.5672	0.1704
type	D	-0.08703	0.3746	29	-0.23	0.8179	0.05	-0.8532	0.6792
type	E	0.3301	0.3046	29	1.08	0.2874	0.05	-0.2928	0.9531

Estimates

Label	Estimate	Standard Error	DF	tValue	Pr>|t|	Alpha	Lower	Upper
E VS. OTHERS	0.6714	0.2675	29	2.51	0.0179	0.05	0.1243	1.2186

The parameter estimates output from `GLIMMIX` is shown below, which is the same as the one from the macro. Because we selected the residual degrees of freedom method with `GLIMMIX`, the significance levels from the macro and procedure are also the same.

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error
year	0.1174	0.1146
period	0.07066	0.1161
year*period	0	.
Residual (VC)	1.6702	0.4690

Solutions for Fixed Effects

Effect	type	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		-5.6799	0.3286	1	-17.28	<.0001
type	E	0.3301	0.3046	29	1.08	0.2874
type	D	-0.08703	0.3746	29	-0.23	0.8179
type	C	-0.6984	0.4248	29	-1.64	0.1110
type	B	-0.5798	0.2277	29	-2.55	0.0164
type	A	0	.	.	.	.

Estimates

Label	Estimate	Standard Error	DF	tValue	Pr>|t|	Alpha	Lower	Upper
E vs. Others	0.6714	0.2675	29	2.51	0.0179	0.05	0.1243	1.2186

As can be seen above and verified on the SAS website, %HPGLIMMIX obtains exactly the same results as both the GLIMMIX and the %GLIMMIX macro. However, with an estimated 0 for the *year* × *period* variance, it is probably advisable to refit the model without this interaction random effect. That is, one can use the following statement in the above code:

```
random year period;
```

4.2. Conditional binomial mixed model

We have already seen that the macro produces the same results as the GLIMMIX procedure using the linearization RSPL method when the same model structure is used. In this example, we fit a hierarchical GLMM to data with an assumed conditional binomial distribution, based on the data sets analyzed in Kriss, Paul, and Madden (2012). The incidence of diseased wheat spikes (heads) in a three-level hierarchy was analyzed: counties, fields nested within counties, and sites nested within fields within counties. The number of diseased (y) and total (n) wheat spikes was determined at each site within each field within each county, and all effects were assumed to be random. A complementary log-log (CLL) link function was used and it was

assumed that y had a conditional binomial distribution. The number of counties is set to 62; this is larger than the number used in the original study, but is useful for showing the advantage of the macro. We used the linearization method to fit a hierarchical GLMM to a simulated data set that is based on typical data, and set of results, in Kriss *et al.* (2012).

Here we emphasize the second key difference between the %HPGLIMMIX macro and the GLIMMIX procedure mentioned in Section 3. When using the %HPGLIMMIX macro to fit a mixed model for a conditional distribution without free scale parameter, there is an extra variance term, the residual variance, that must be fixed at 1. GLIMMIX, however, automatically handles this. So, with this example, there are three variance terms with the procedure and four with the macro (although the last one is held at 1).

```
data work.plant;
        CALL STREAMINIT(9873123);
        do sim = 1 to 1;
             inter = -2;
             n = 50;
             do county = 1 to 62;
                 varc = 0.65;
                 uc = rand('normal')*sqrt(varc);
                 do field = 1 to 10;
                      varf = .50;
                      uf = rand('normal')*sqrt(varf);
                      do site = 1 to 20;
                           vars = .07;
                           us = rand('normal')*sqrt(vars);
                           eta = inter + uc + uf + us;
                           p = (1-exp(-exp(eta)));
                           y = rand('binomial',p,n);
                           output;
                      end;
                end;
           end;
     end;
run;
```

The following log pieces show the code and resource usage from the GLIMMIX procedure and the %HPGLIMMIX macro, respectively, for estimation and comparison purposes. Note that in the macro, the PARMS statement is used to not only specify the starting value for variance (or more generally, the variance-covariance) parameters, but also fixes the last (4th in this example) variance to be 1 using the HOLD=4 option. Because random effects have a nested structure, we specified the SUBJECT= option in the RANDOM statement to process the data by subject in order to make the computing more efficient. Note that both GLIMMIX and %HPGLIMMIX support this option in the RANDOM statement.

```
2442  title 'PQL ("Penalized Quasi-Likelihood"), 3-level hierarchy, scale=1';
2443  title2 'int=-2, vars: 0.65, .50, .07 [n=50], 16 counties';
2444  /* As described in GLIMMIX manual, with discrete distributions,
```

```
2445      the residual scale is automatically 1. */
2446  proc glimmix data=work.plant ;
2447      class county field site;
2448      model y/n =  / dist=binomial link=cll  s ;
2449      random int field site(field) /subject=county;
2450      nloptions maxiter=100 tech=QUANEW;
2451      ods output CovParms=Cov_glimmix;
2452  run;

NOTE: Convergence criterion (PCONV=1.11022E-8) satisfied.
NOTE: The data set WORK.COV_GLIMMIX has 3 observations and 4 variables.
NOTE: PROCEDURE GLIMMIX used (Total process time):
      real time           5:51.19
      user cpu time       5:47.75
      system cpu time     1.98 seconds
      memory              672046.84k
      OS Memory           688696.00k
      Timestamp           08/13/2013 06:05:00 PM

2453
2454
2455
2456  /*  Now use new %HPGLIMMIX, also with one more level of
2457       of variation (residual), HELD at 1.   */
2458
2459  ods select ParmSearch CovParms ParameterEstimates FitStatistics;
2460  title3 'using new %hpglmmix, with fixed residual=1';
2461  %hpglimmix(data =work.Plant,
2462          stmts=%str(
2463              class county field site;
2464              model y/n =  /  s ;
2465              random int field site(field) /subject=county;
2466              parms (.6) (.5) (.1) (1) / hold=4; *<-- fix "residual" at 1;
2467          ),
2468      error=binomial, maxit=50,
2469      tech=QUANEW,
2470      link=cloglog
2471  );

       The HPGLIMMIX Macro

Data Set            : WORK.PLANT
Error Distribution  : BINOMIAL
Link Function       : COMPLEMENTARY LOG LOG
Response Variable   : Y/N
```

```
Job Starts at : 13AUG2013:18:05:00
    HPGLIMMIX Iteration History

Iteration    Convergence criterion
    1            0.0969208415  12 sec
    2            0.0493967059  11 sec
    3            0.0036269351  10 sec
    4            0.0001263753  12 sec
    5            0.0000110672  10 sec
    6            1.3960177E-6  18 sec
    7            3.4386804E-9  15 sec

Output from final Proc HPMixed run:
Job Ends at : 13AUG2013:18:07:03
2472  run;
2473  title;title2;title3;
```

Although the output is not shown, the same variance parameter estimates were obtained with the macro and the procedure. The `GLIMMIX` procedure took slightly more than 5 minutes 51 seconds to converge and the `%HPGLIMMIX` macro took about 2 minutes 3 seconds. For random effects with nested structure, using the `SUBJECT=` option to enable processing by subject is highly encouraged. If the random statement is specified as `RANDOM county field(county) site(field county)` , the `GLIMMIX` procedure would take several hours to finish, whereas the macro would take less than an hour to finish. In situations where the `GLIMMIX` procedure and the `%HPGLIMMIX` macro have a similar performance, the `GLIMMIX` procedure should be preferred since it provides a much wider range of features and covariance types.

4.3. Reduction in running time and memory requirement for large scale GLMM

In this example, we fit a GLMM to the simulated microarray data from Example 45.4 in SAS Institute Inc. (2011b) for the `HPMIXED` procedure, but assume a gamma distribution (conditional on the random effects) instead of a normal conditional distribution as in SAS Institute Inc. (2011b). The purpose is to push the scale of model to a higher limit and demonstrate the great advantage of using the macro instead of the procedure for such big data problems. The data set simulates a so-called loop microarray design structure, which is commonly used in such studies. There are 500 genes and 6 treatments, each gene occurs in 6 arrays, and each array has 2 dyes; so-called pins and dips on the arrays give multiple observations. The model assumes the same structure as in SAS Institute Inc. (2011b), which is also described as case study 16.12 in Littell, Milliken, Stroup, Wolfinger, and Schabenberger (2006). Fixed effects are: gene, treatment, dye, gene-treatment interaction, dye-gene interaction, and array pin; random effects are array, array-gene interaction, dip-within-array, array-pin interaction. This is a large model with 4513 columns in the design matrix of fixed effects and 3054 columns in the design matrix of random effects, which makes the mixed model equation having more than 7500 columns in total, with a sparsity of only 0.14537%. The data generation is given in the SAS program, and is the same as found in SAS Institute Inc. (2011b), except that η is a linear function of the fixed and random effects, and that the response variable has a

conditional gamma distribution with expected value $\exp(\eta)$ and scale parameter of 0.5 (which was estimated in the model fitting).

The following example used both the GLIMMIX procedure and the %HPGLIMMIX macro to estimate the same model and showed the difference in time consumption and memory usage. Results were stored using the ODS output system, and additional code for a data step were written to compare key results side-by-side in the SAS log (which is displayed). As shown below, on a Windows PC equipped with Intel i5-3570K CPU running at 3.8GHz, the macro took a total of about 67 minutes to finish, while the GLIMMIX procedure took more than 45 hours, a 40-plus folds saving in time. As an aside, we attempted to fit the GLMM using the Laplace (likelihood approximation) method of the GLIMMIX procedure, but convergence could not be obtained after 5 days (unpublished).The following log shows the input program code and information on the model fitting with the linearization method, as well as the execution time, and a comparison ot the estimates of variance parameters from both the macro and the procedure. Note that the TEST statement is used in the macro to perform an F test of the treatment effect with the macro. Also note that the Newton-Raphson with ridging (NRRIDG) was explicitly chosen for the inner-iteration optimization technique; although it is the default with the macro, it is shown here to demonstrate its use. As can be seen, the variance parameter results are identical up to 8 decimal places. The actual ODS output in the results window are not shown to save space.

```
5                The SAS System           22:03 Friday, December 28, 2012

1380
1381    %hpglimmix(data=work.microarrayG,
1382                stmts=%str(
1383                  class marray dye trt gene pin dip;
1384                    model response = dye trt gene dye*gene trt*gene pin;
1385                    random marray marray*gene dip(marray) pin*marray;
1386                    test trt;
1387                   ),
1388                error=gamma,
1389                link=log,
1390                tech=NRRIDG
1391    );

       The HPGLIMMIX Macro

Data Set            : WORK.MICROARRAYG
Error Distribution  : GAMMA
Link Function       : LOG
Response Variable   : RESPONSE


Job Starts at : 28DEC2012:22:03:36
    HPGLIMMIX Iteration History

Iteration    Convergence criterion
```

```
 1            2             118 sec
 2            2             117 sec
 3            2             116 sec
 4            2             116 sec
 5            2             117 sec
 6            0.2765015635  117 sec
 7            0.1399609947  122 sec
 8            0.0666621139  120 sec
 9            0.0325548942  117 sec
10            0.0159305151   117 sec
11            0.0077325255   120 sec
12            0.0038101692   118 sec
13            0.0018487198   120 sec
14            0.00091249     119 sec
15            0.0004429375   120 sec
16            0.0002189407   119 sec
17            0.000106098    120 sec
18            0.0000518751   118 sec
19            0.0000251754   120 sec
20            0.0000123817   117 sec
21            5.952897E-6    120 sec
22            2.9335583E-6   120 sec
23            1.4340711E-6   120 sec
24            7.0181653E-7   120 sec
25            3.4342231E-7   120 sec
26            1.6738384E-7   120 sec
27            8.1505387E-8   119 sec
28            3.8524046E-8   119 sec
29            2.9539118E-8   120 sec
30            1.8899534E-8   120 sec
31            8.2400631E-9   119 sec

Output from final Proc HPMixed run:
Job Ends at : 28DEC2012:23:10:02
1392
1393
1394       options notes source;
1395       *ods select none;
1396
1397       ods output ParameterEstimates =beta_glimmix;
1398       ods output CovParms = cov_glimmix;
1399       proc glimmix data=work.microarrayG   ;
1400          class marray dye trt gene pin dip;
1401          model  response = dye trt gene dye*gene trt*gene pin
1402                                              / dist=gamma link=log s;
1403          random marray marray*gene dip(marray) pin*marray;
1404          nloptions tech=NRRIDG  maxiter=50;
```

```
1405       run;

NOTE: Convergence criterion (PCONV=1.11022E-8) satisfied.
NOTE: The data set WORK.COV_GLIMMIX has 5 observations and 3 variables.
NOTE: The data set WORK.BETA_GLIMMIX has 4513 observations and 10 variables.
NOTE: The PROCEDURE GLIMMIX printed pages 210-314.
NOTE: PROCEDURE GLIMMIX used (Total process time):
      real time           45:14:43.46
      user cpu time       45:03:50.86
      system cpu time     1:29.76
      memory              1888794.93k
      OS Memory           1898292.00k
      Timestamp           12/30/2012 08:24:45 PM

1406       ods select all;
1407
1408       /* Compare results */
1409       data _null_;
1410            set _cov;
1411                put  CovParm=;
1412                put @1  '%HPGLIMMIX  '  Estimate=  best10.9 @@ ;
1413                set cov_glimmix;
1414                put @40 'GLIMMIX     '  Estimate=  best10.9;
1415                put ;
1416       run;
7              The SAS System          22:03 Friday, December 28, 2012

CovParm=MArray
%HPGLIMMIX  Estimate=0.00117475        GLIMMIX     Estimate=0.00117475

CovParm=MArray*Gene
%HPGLIMMIX  Estimate=0.00277636        GLIMMIX     Estimate=0.00277636

CovParm=Dip(MArray)
%HPGLIMMIX  Estimate=0.00170539        GLIMMIX     Estimate=0.00170539

CovParm=MArray*Pin
%HPGLIMMIX  Estimate=0.03140975        GLIMMIX     Estimate=0.03140975

CovParm=Residual
%HPGLIMMIX  Estimate=0.48076186        GLIMMIX     Estimate=0.48076186
NOTE: There were 5 observations read from the data set WORK._COV.
NOTE: There were 5 observations read from the data set WORK.COV_GLIMMIX.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      user cpu time       0.00 seconds
      system cpu time     0.00 seconds
```

```
memory              313.71k
OS Memory           11044.00k
Timestamp           12/30/2012 08:24:45 PM
```

5. Conclusion

The %HPGLIMMIX macro, based on the %GLIMMIX macro of SAS, provides a convenient way to fit GLMMs to large-scale data sets with large numbers of fixed or random effects. Depending on the size and sparseness of the design matrices, considerable time and memory savings can result, relative to the use of the GLIMMIX procedure. The macro is based strictly on the use of the doubly iterative linearization method, which is a very general method that can be applied to a wide range of GLMMs. Although the parameter estimates may be more biased than found with the likelihood approximation methods, these latter approaches are not computationally well suited to large-scale problems at this time. The bias problem has been shown by Stroup (2012) to be an issue with discrete data only under extreme conditions, such as with very small number of trials per sampling unit.

On the other hand, the %HPGLIMMIX macro is built on the HPMIXED procedure, hence the limitations of this procedure apply. It is designed for special cases of a mixed model with large but sparse design matrix and only a few distributions are supported. For GLMMs with large but dense design matrices, the performance of this macro will be worse than that of the GLIMMIX procedure. In addition, only a subset of covariance structures of the GLIMMIX procedure are available as of this writing. Some other limitations include type 3 test results are not provided by default because dense matrix computation is involved, and degrees of freedom methods such as the Kenward-Roger method and the Satterthwaite method are not supported because they require to store and operate on the dense mixed model equation.

Therefore, users that need those features will have to use the GLIMMIX procedure. However, they can use the %HPGLIMMIX macro for large scale (big data) problems and to accelerate the GLIMMIX procedure analyses for very large problems. The idea is to maximize the likelihood and produce parameter estimates more quickly using the %HPGLIMMIX macro, and then to pass these parameter estimates to the GLIMMIX procedure for some further analysis that is not available within the %HPGLIMMIX macro, see Example 45.3 in SAS Institute Inc. (2011a) for details.

References

Antonio K, Beirlant J (2007). "Actuarial Statistics with Generalized Linear Mixed Models." *Insurance: Mathematics and Economics*, **40**(1), 58–76.

Bates D, Maechler M, Bolker B, Walker S (2014). ***lme4****: Linear Mixed-Effects Models Using* ***Eigen*** *and S4*. R package version 1.1-6, URL http://CRAN.R-project.org/package=lme4.

Breslow NE, Clayton DG (1993). "Approximate Inference in Generalized Linear Mixed Models." *Journal of the American Statistical Association*, **88**(421), 9–25.

Frees EW, Young VR, Lou Y (1999). "A Longitudinal Data Analysis Interpretation of Credibility Models." *Insurance: Mathematics and Economics*, **24**(3), 229–247.

Gilmour AR, Thompson R, Cullis BR (1995). "Average Information REML: An Efficient Algorithm for Variance Parameter Estimation in Linear Mixed Models." *Biometrics*, **51**(4), 1440–1450.

Kaas R, Dannenburg D, Goovaerts M (1997). "Exact Credibility for Weighted Observations." *ASTIN Bulletin*, **27**(2), 287–295.

Kerr MK, Martin M, Churchill GA (2000). "Analysis of Variance for Gene Expression Microarray Data." *Journal of Computational Biology*, **7**(6), 819–837.

Kriss AB, Paul PA, Madden LV (2012). "Characterizing Heterogeneity of Disease Incidence in a Spatial Hierarchy: A Case Study from a Decade of Observations of Fusarium Head Blight of Wheat." *Phytopathology*, **102**(9), 867–877.

Littell R, Milliken G, Stroup W, Wolfinger R, Schabenberger O (2006). *SAS System for Mixed Models*. SAS Institute, Inc., 2nd edition.

Maiti T (2001). "Robust Generalized Linear Mixed Models for Small Area Estimation." *Journal of Statistical Planning and Inference*, **98**(1–2), 225–238.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. 2nd edition. Chapman and Hall, London.

Milsom TP, Langton SD, Parkin WK, Peel S, Bishop JD, Hart JD, Moore NP (2000). "Habitat Models of Bird Species Distribution: An Aid to the Management of Coastal Grazing Marshes." *Journal of Applied Ecology*, **37**(5), 706–727.

Pinheiro JC, Bates DM (1995). "Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model." *Journal of Computational and Graphical Statistics*, **4**(1), 12–35.

Pinheiro JC, Chao EC (2006). "Efficient Laplacian and Adaptive Gaussian Quadrature Algorithms for Multilevel Generalized Linear Mixed Models." *Journal of Computational and Graphical Statistics*, **15**(1), 58–81.

SAS Institute Inc (2007). *SAS Knowledge Base Sample 25030:* `%GLIMMIX` *Macro to Fit the Generalized Linear Mixed Model*. SAS Institute Inc. [Online; accessed 2012-12-19], URL http://support.sas.com/kb/25/030.html.

Raudenbush S, Yang ML, Yosef M (2000). "Maximum Likelihood for Generalized Linear Models with Nested Random Effects via High-Order, Multivariate Laplace Approximation." *Journal of Computational and Graphical Statistics*, **9**(1), 141–157.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

SAS Institute Inc (2011a). *The SAS System, Version 9.3*. SAS Institute Inc., Cary, NC. URL http://www.sas.com/.

SAS Institute Inc (2011b). *SAS/IML 9.3 User's Guide*. SAS Institute Inc., Cary, NC. URL http://www.sas.com/.

Schabenberger O, Pierce F (2002). *Contemporary Statistical Models for the Plant and Soil Sciences*. CRC Press.

Schall R (1991). "Estimation in Generalized Linear Models with Random Effects." *Biometrika*, **78**(4), 719–727.

Stroup W (2012). *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*. Texts in Statistical Science. Chapman & Hall/CRC.

Vergara P, Aguirre I J, Fernandez-Cruz M (2007). "Arrival Date, Age and Breeding Success in White Stork Ciconia Ciconia." *Journal of Avian Biology*, **38**(5), 573–579.

Wolfinger R (1993). "Laplace's Approximation for Nonlinear Mixed Models." *Biometrika*, **80**(4), 791–795.

Wolfinger R, O'Connell M (1993). "Generalized Linear Mixed Models: A Pseudo-Likelihood Approach." *Journal of Statistical Computation and Simulation*, **48**(3–4), 233–243.

Wolfinger RD, Gibson G, Wolfinger ED, Bennett L, Hamadeh H, Bushel P, Afshari C, Paules RS (2001). "Assessing Gene Significance from cDNA Microarray Expression Data via Mixed Models." *Journal of Computational Biology*, **8**(6), 625–637.

Affiliation:

Liang Xie
Microsoft Corp.
City Center Plaza
Bellevue, WA, United States of America
E-mail: xie1978@hotmail.com
URL: http://sas-programming.blogspot.com/

Laurence V. Madden
Ohio State University
Department of Plant Pathology
Wooster, OH, United States of America
E-mail: madden.1@osu.edu
URL: http://plantpath.osu.edu/madden/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 58, Issue 8 *Submitted:* 2013-03-22
June 2014 *Accepted:* 2014-02-19