

Reimplementation of Zhang *et al.* 2011

Ben Bolker, Reinhold Kliegl, Dave Fournier

November 9, 2011

1 Introduction

This is a reconstruction of (Zhang et al., 2011). The purpose is to check the results and to try to investigate a wider range of R packages implementing GLMMs. **This is a work in progress! I haven't checked any of this with the authors of the original paper yet.**

We used R Under development (unstable) (2011-10-06 r57181) and package versions:

<code>coefplot2</code>	<code>glmmADMB</code>	<code>glmmML</code>	<code>lme4</code>	<code>lme4a</code>	<code>lme4Eigen</code>
0.1.1	0.6.4	0.82-1	0.999375-42	0.9996875-1	0.9996875-5
<code>npmlreg</code>	<code>repeated</code>	<code>sabreR</code>			
0.44	1.0	1.1			

The `coefplot2` and `glmmADMB` packages must be installed from <http://r-forge.r-project.org>; `repeated` and its dependency `rmutil` must be installed from Jim Lindsey's R code page (<http://www.commanster.eu/rcode.html>); all others are on CRAN.

- R packages/functions that can do the respiratory data example (binomial, single grouping variable) using a deterministic (PQL, Laplace or GHQ) algorithm: `lme4::glmer` (and the development `lme4a`, `lme4Eigen` versions); `glmmADMB/AD Model Builder`; `glmmML`; `sabreR`; `repeated`; `MASS::glmmPQL`, (ASREML), `npmlreg`, `gamlss.mx` (note: `gamlss.mx`'s documentation says that it is built on functions from `npmlreg`, so the results are likely to be the same), ...
- Packages that can do the simulation (binomial, single grouping variable, multiple responses (random-slope model): `lme4`, `glmmADMB/AD Model Builder`, `MASS::glmmPQL`, ...?

To do:

- respiratory example:
 - understand basic divergence of current results from Zhang et al. (esp. intercept) — contrast difference?? Did Zhang use `visit` at all? They say:

We also applied the different procedures/packages to a multi-center randomized clinical trial comparing two treatments for a respiratory illness [23]. There were 111 subjects randomized to active (54) or placebo (57) treatment across two sites. The primary outcome was respiratory status (0 = poor, 1 = good), which was assessed at four visits. The study data were analyzed by Davis using distribution-free (or semi-parametric) longitudinal methods [23]. In addition to the treatment conditions, he also included baseline respiratory status, study site, sex and age as covariates. There was no missing data in the outcome, the predictor (treatment conditions), and the covariates. Our reanalysis using GLMM included the same predictor and covariates as the fixed effects, and a random intercept.

- comment on marginal vs conditional parameters (e.g. cite Casella) – difference between glm/GEE and mixed-model results
- nag Zhang et al. about code?
- simulations:
 - implement other packages (glmmML, development lme4, etc.)
 - finish up ADMB stuff: get GH5, importance sampling results
 - compare glmmADMB with raw ADMB results — should be identical?
- general:
 - fix accessory functions (`coefstab` etc.) for development lme4 versions
 - consider MCMC-based solutions (MCMCglmm, INLA, JAGS/BUGS)

2 Estimates from data

The data that Zhang *et al.* use (originally from (Davis, 1991)) can be found in the `geepack` package as `respiratory`, and in the `HSAUR` package. The version in the `HSAUR` package differs from the original ones in two respects:

1. male and female labels are exchanged
2. for record 428, the outcome/status should be “poor” (0), not “good” (1)

```
> library(geepack)
> data(respiratory)
```

Because `id` is implicitly nested in `center`, we’ll use the interaction of the two as the blocking variable (`center` will be included in the model as a fixed effect).

```

> respiratory <- transform(respiratory,
                           center=factor(center),
                           id=factor(id),
                           idctr=interaction(id,center),
                           treat=factor(treat,levels=c("P","A")),
                           sex = factor(sex,levels=c("M","F")))

(Davis, 1991) estimates:

> indep.est <- c(Treatment=1.265, Centre=0.649,
                 Sex=0.137, Age=-0.019, Baseline=1.846)
> indep.se <- c(Treatment=0.347, Centre=0.353,
                Sex=0.440, Age=-0.013, Baseline=0.346)

> summary(m_glm <- glm(outcome ~ treat + center + sex + age + baseline,
                       family="binomial", data=respiratory))

Call:
glm(formula = outcome ~ treat + center + sex + age + baseline,
    family = "binomial", data = respiratory)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2906  -0.8610   0.4402   0.9041   1.9164

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.856106   0.335142  -2.554  0.01064 *
treatA       1.265356   0.235000   5.384 7.27e-08 ***
center2      0.649490   0.238258   2.726  0.00641 **
sexF         0.136780   0.293330   0.466  0.64100
age         -0.018756   0.008816  -2.128  0.03337 *
baseline     1.845720   0.239267   7.714 1.22e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 609.41  on 443  degrees of freedom
Residual deviance: 487.49  on 438  degrees of freedom
AIC: 499.49

Number of Fisher Scoring iterations: 4

>
> library(coefplot2)

```

2.1 glmmML

```
> library(glmmML)

> m_glmmML.L <- glmmML(outcome~center+treat+sex+age+baseline,
                        cluster=idctr,
                        family=binomial,data=respiratory,
                        method="Laplace")
> m_glmmML.GHQ8 <- glmmML(outcome~center+treat+sex+age+baseline,
                           cluster=idctr,
                           family=binomial,data=respiratory,
                           method="ghq")

> add_ctab(m_glmmML.L)
> add_ctab(m_glmmML.GHQ8)
> detach("package:glmmML")
```

2.2 repeated

```
> library(repeated)

> m_repeated.L <- glmm(outcome~center+treat+sex+age+baseline,
                       nest=idctr,
                       family=binomial,data=respiratory,points=1)
> m_repeated.GHQ8 <- glmm(outcome~center+treat+sex+age+baseline,
                           nest=idctr,
                           family=binomial,data=respiratory,points=8)

> add_ctab(m_repeated.L)
> add_ctab(m_repeated.GHQ8)
> detach("package:repeated")
```

2.3 MASS::glmmPQL

```
> library(MASS)

> m_glmmPQL <- glmmPQL(outcome~center+treat+sex+age+baseline,
                       random=~1|idctr,
                       family=binomial,data=respiratory)

> add_ctab(m_glmmPQL)
```

2.4 lme4

2.4.1 stable version

```
> library(lme4)
```

```

> m_glmer_4.L <- glmer(outcome~center+treat+sex+age+baseline+(1|idctr),
  family=binomial,data=respiratory)
> m_glmer_4.GHQ8 <- glmer(outcome~center+treat+sex+age+baseline+(1|idctr),
  family=binomial,data=respiratory,nAGQ=8)

> add_ctab(m_glmer_4.L)
> add_ctab(m_glmer_4.GHQ8)
> detach("package:coefplot2",unload=TRUE)
> detach("package:lme4",unload=TRUE)

```

2.4.2 development 1 (lme4a)

```
> library(lme4a)
```

AGHQ is not implemented yet for lme4a.

```
> m_glmer_4a.L_fit <- glmer(outcome~center+treat+sex+age+baseline+(1|idctr),
  family=binomial,data=respiratory)
```

FIXME: coeftab doesn't work for lme4a yet.

2.4.3 development 2 (lme4Eigen)

```
> library(lme4Eigen)
```

Note that lme4Eigen is truly bleeding-edge — at the moment it's crashing R (!) so not included. AGHQ is not implemented yet for lme4Eigen.

```
> m_glmer_4e.L_fit <- glmer(outcome~center+treat+sex+age+baseline+(1|idctr),
  family=binomial,data=respiratory)
```

```
> m_glmer_4e.L <- gg(m_glmer_4e.L_fit)
> add_ctab(m_glmer_4e.L,raw=TRUE)
> detach("package:lme4Eigen",unload=TRUE)

```

2.5 glmmADMB

```
> library(glmmADMB)
```

```
> m_glmmADMB.base <- glmmadmb(outcome~center+treat+sex+age+baseline,
  random=~1|idctr,
  family="binomial",data=respiratory)
> m_glmmADMB.is100 <- glmmadmb(outcome~center+treat+sex+age+baseline,
  random=~1|idctr,
  family="binomial",
  admb.opts=admbControl(impSamp=100),
  data=respiratory)
> m_glmmADMB.is200 <- glmmadmb(outcome~center+treat+sex+age+baseline,
```

```

                                random=~1|idctr,
                                family="binomial",
                                admb.opts=admbControl(impSamp=200),
                                data=respiratory)
>
> add_ctab(m_glmADMB.base)
> add_ctab(m_glmADMB.is100)
> add_ctab(m_glmADMB.is200)
> detach("package:glmmADMB")

```

2.6 sabreR

```

> library(sabreR)

> resp <- respiratory
> resp$idctr <- as.integer(resp$idctr)
> attach(resp)
> sabreRfit <- sabre(outcome~center+treat+sex+age+baseline,
                    case=idctr)
> detach(resp)

> add_ctab(m_sabreR,raw=TRUE)
> detach("package:sabreR")

```

2.7 npmlreg

```

> library(npmlreg)

> m_npmlreg.GHQ8 <- allvc(outcome~center+treat+sex+age+baseline,
                        random=~1|idctr,
                        family=binomial,
                        k=8,
                        random.distribution="gq",
                        data=respiratory)

```

```

[1] "1"      "idctr"
1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..
EM algorithm met convergence criteria at iteration # 14
Disparity trend plotted.

```

```

> add_ctab(m_npmlreg.GHQ8)

```

```

Call: allvc(formula = outcome ~ center + treat + sex + age + baseline, random = ~1 | i

```

Coefficients:

```

      Estimate Std. Error  t value

```

```

(Intercept) -1.58979620 0.44677386 -3.558391
center2      0.84482938 0.30757906  2.746706
treatA       2.12157775 0.32526162  6.522681
sexF         0.31329489 0.37609933  0.833011
age         -0.02466708 0.01160836 -2.124941
baseline     2.88461368 0.35075178  8.224089
z            1.94282415 0.20643945  9.411109

```

```
Random effect distribution - standard deviation:          1.942824
```

```
-2 log L:          432.6          Convergence at iteration 14
```

```

> library(INLA)
> respiratory$N <- 1
> ## inla(outcome~center+treat+sex+age+baseline+f(idctr,model="iid"),
> ##      family="binomial",data=respiratory,Ntrials=N)
> detach("package:INLA")
> save("ctablist",file="Zhang_reanalysis.RData")

```

2.8 Summary

Here are the combined estimates. None of the packages I've tried here diverge particularly far from each other, but none of them come anywhere close to the lme4 values reported by Zhang et al. (package difference) ... the closest match is with Zhang et al.'s reported glmmML estimates, which are fairly close to the R estimates with the exception of the intercept (way off) and the Treatment effect (off by more than rounding error, but only about 3%) — the rest are off by less than 1% (except for Sex, which is probably a rounding issue).

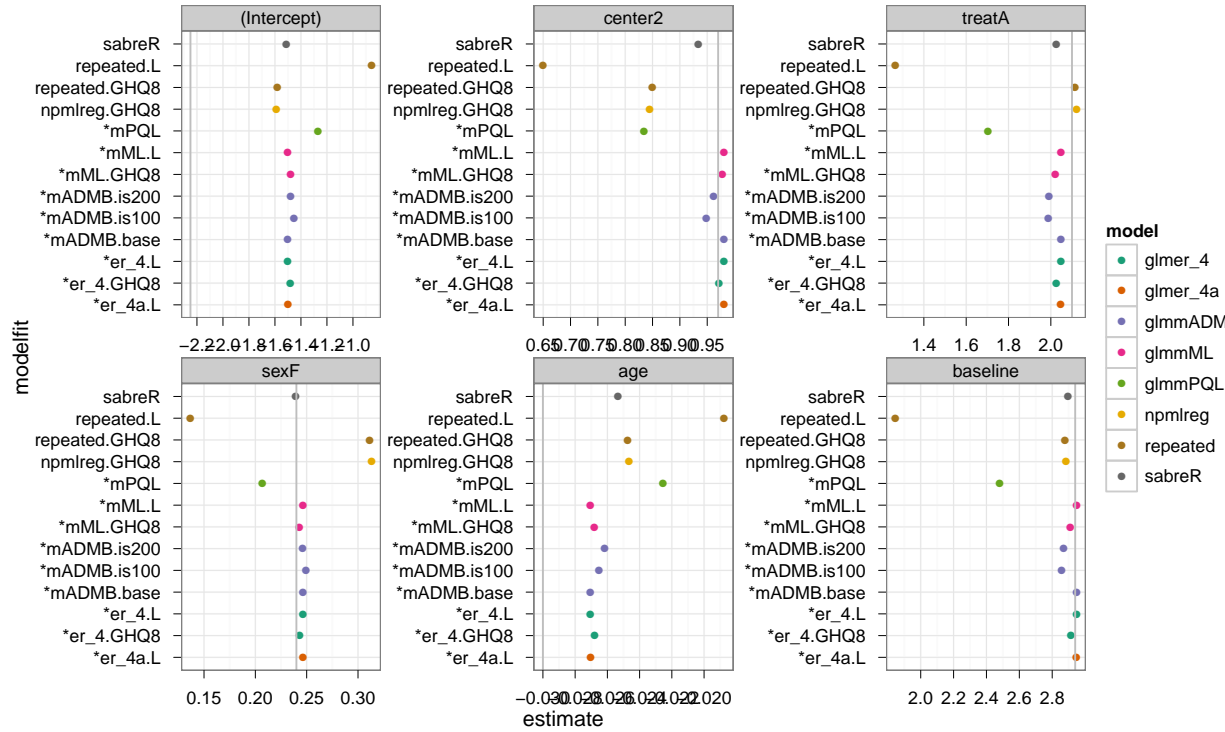
	Intercept	Center	Treatment	Sex	Age	Baseline
Zhang_lme4_GH	2.4400000	0.9200000	-1.970000	-0.2800000	-0.0300000	2.850000
Zhang_glmmML	1.7000000	0.8500000	-1.740000	-0.0800000	-0.0200000	2.010000
Zhang_NLMIXED	2.2500000	0.9700000	-2.100000	-0.2400000	-0.0300000	2.940000
m_glmmML.L	-1.5012951	0.9809060	2.048374	0.2464449	-0.02705820	2.948312
m_glmmML.GHQ8	-1.4789769	0.9776592	2.021241	0.2427480	-0.02681527	2.910442
m_repeated.L	-0.8561061	0.6494901	1.265355	0.1367804	-0.01875635	1.845719
m_repeated.GHQ8	-1.5806556	0.8498820	2.113674	0.3112463	-0.02473351	2.877415
m_glmmPQL	-1.2679104	0.8343282	1.702316	0.2069005	-0.02254507	2.480673
m_glmer_4.L	-1.5012775	0.9809104	2.048357	0.2464680	-0.02705861	2.948231
m_glmer_4.GHQ8	-1.4823065	0.9713807	2.026090	0.2432837	-0.02678836	2.913547
m_glmer_4a.L	-1.5006865	0.9804209	2.047024	0.2464219	-0.02703852	2.946716
m_glmmADMB.base	-1.5013000	0.9809100	2.048400	0.2464700	-0.02705900	2.948300
m_glmmADMB.is100	-1.4534000	0.9482000	1.988000	0.2492600	-0.02651900	2.858500
m_glmmADMB.is200	-1.4780000	0.9620300	1.991400	0.2461000	-0.02615900	2.868900

```

m_sabreR          -1.5130000 0.9337600 2.026300 0.2392600 -0.02534000 2.895600
m_npmlreg.GHQ8   -1.5897962 0.8448294 2.121578 0.3132949 -0.02466708 2.884614

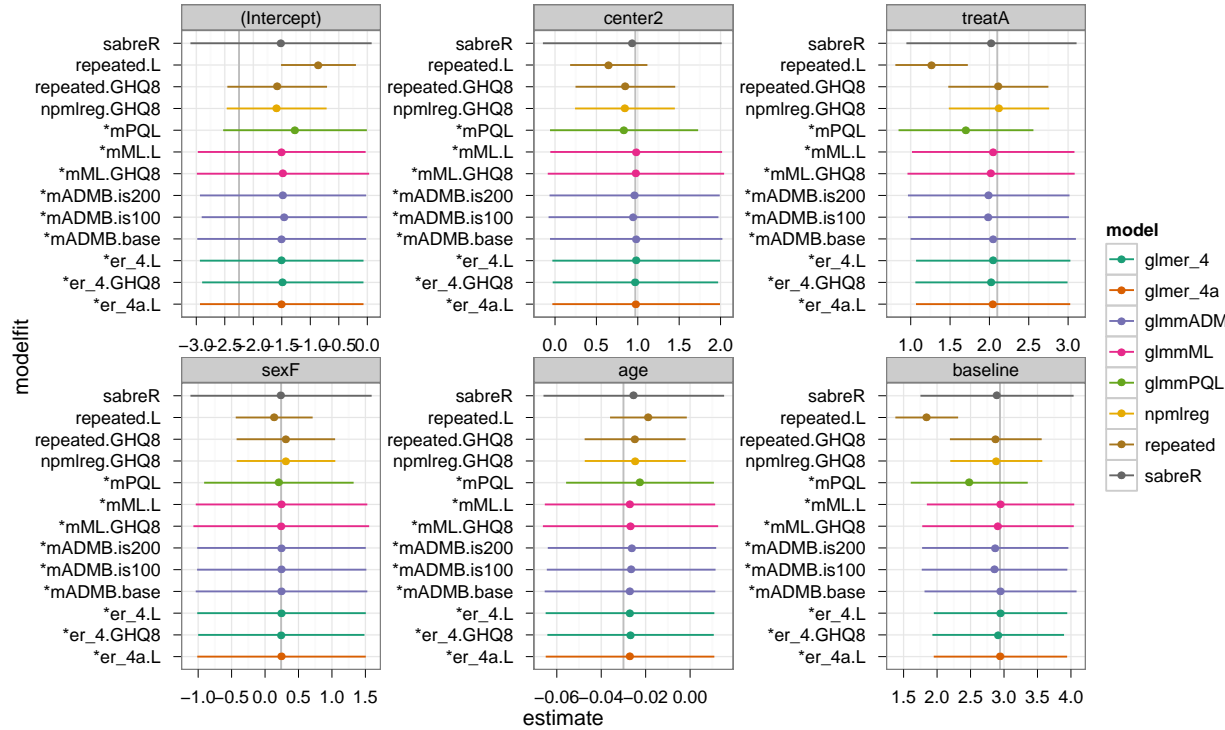
```

First a plot with just the estimates:



To do: Make x axes prettier!

Now a plot including estimated standard errors:



Conclusion: for this particular data set, results generally agree pretty well.

- Except for the intercept, the estimates are all reasonably close to Zhang’s estimates (with sign corrections as appropriate), although in general the R estimates based on Laplace/GHQ almost all deviate from Zhang’s estimates consistently (this is clearer in the picture without error bars)
- glmmADMB, glmmML, and lme4 all agree quite closely (at least considering the scale of the uncertainty), both in estimates and confidence intervals, regardless of whether Laplace, L+importance sampling, or GHQ is used (lme4 has slightly smaller standard error/confidence interval estimates for the baseline parameter)
- sabreR is pretty close too, but a little farther off, and has slightly wider confidence intervals
- glmmPQL deviates, as expected.
- Surprisingly, npmlreg and repeated agree with each other and disagree with the (otherwise) consensus. Are they doing something similar like non-adaptive GHQ?? Their confidence intervals are also smaller than everyone else’s ...

3 Simulations

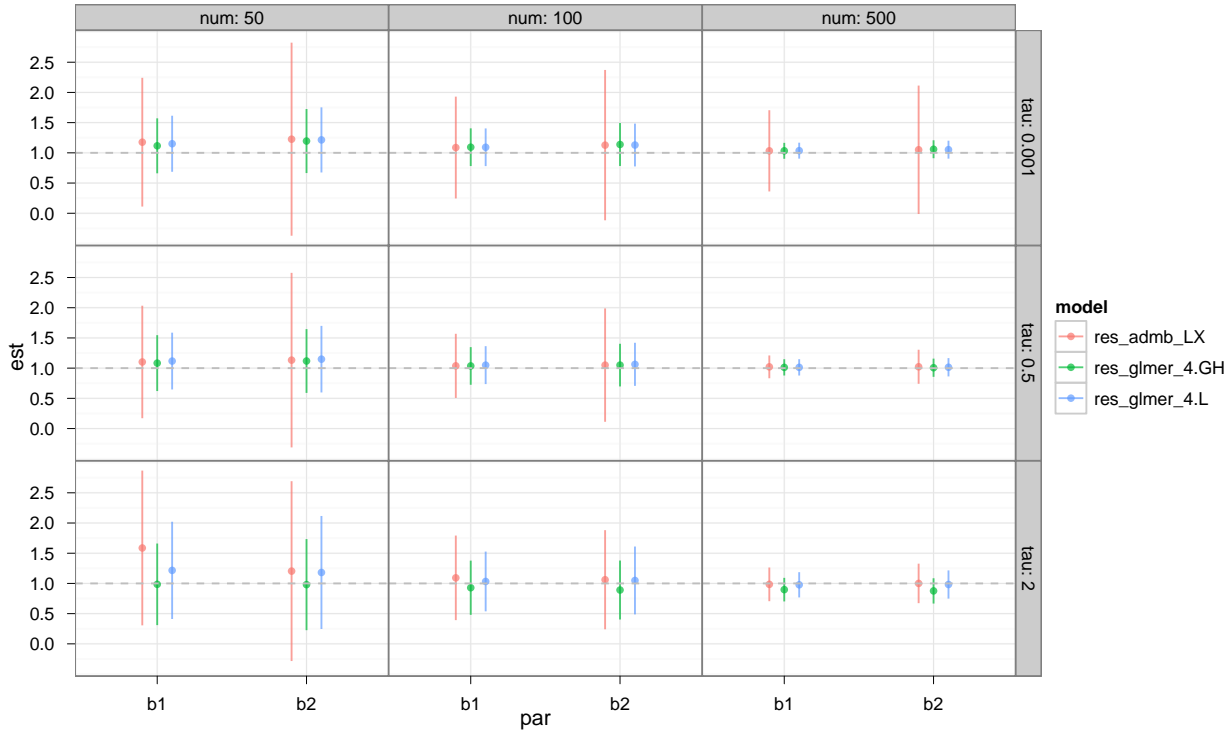
Here is the basic simulation function:

```
> simfun <- function(n,beta=c(1,1),tau=0.001,
                    Cor=matrix(c(1,0.25,0.25,1),nrow=2)) {
  D <- expand.grid(id=factor(1:n),t=1:3)
  D$x <- rnorm(3*n) # x_{it} ~ N(0,1)
  X <- model.matrix(~x,data=D)
  Z <- model.matrix(~id-1+id:x,data=D)
  b <- MASS::mvrnorm(n,mu=c(0,0),Sigma=tau^2*Cor)
  D$y <- rbinom(nrow(D),
               prob=plogis(X %>% beta + Z %>% c(b)),
               size=1)
  D
}
> tauvec <- c(0.001,0.5,2)
> nvec <- c(50,100,500)
```

The simulations themselves are now being done in batch files.

3.1 Mean estimates and CIs

(Note these are the *mean* positions of the upper/lower CIs)



The bias is similar across the board, but ADMB estimates systematically wider confidence intervals (which as shown below are more correct in terms of coverage).

To do: Incorporate IS and GHQ results for ADMB: according to DF these give even better coverage

3.2 Coverage

This is the same as Zhang et al’s “type I error” — I think the term “coverage” is more useful, although technically Zhang et al. are correct — if we treated $\beta_i = (\text{true value})$ as a null hypothesis, the type I error would be equivalent to the coverage ...

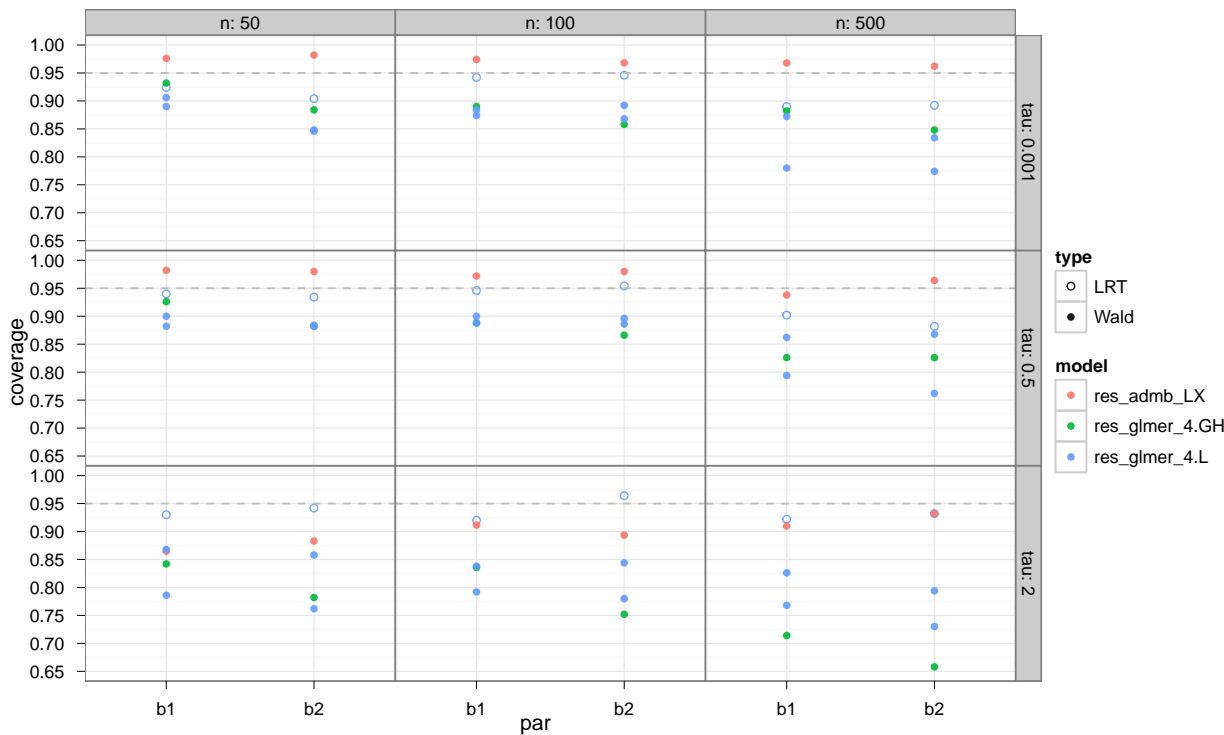
I wanted to try LRT coverage as well as Wald coverage. Since we don’t have the capability to compute profile confidence intervals for GLMMs in `glmer` (nor [?] in any other package that I know of; it’s not implemented in ADMB for RE models either, although one could always do it by brute force/looping), we cheat a bit by fitting a model with the true value set as an offset, and the relevant parameter taken out of the estimation model. (Should explain this better.) The distribution of LRT p values computed in this way (by comparing the full model to the restricted model) is equivalent to the profile likelihood interval coverage. (This might be worth doing in any case, because “test whether the true value

is in the confidence interval” takes less work than “construct the confidence intervals, then see whether they include the true value”.)

The R code for this is as follows:

```
> dat <- simfun(n=500,tau=2)
> library(lme4)
> g0 <- glmer(y~x+(x|id),data=dat,family=binomial)
> dat$int <- 1
> g1 <- glmer(y~0+x+(x|id)+offset(int),data=dat,family=binomial)
> g2 <- glmer(y~1+(x|id)+offset(x),data=dat,family=binomial)
> b1_LRT <- anova(g0,g1)[["Pr(>Chisq)"]][2] ## test of b1=1
> b2_LRT <- anova(g0,g2)[["Pr(>Chisq)"]][2] ## test of b2=1
> coef0 <- fixef(g0)
> stder0 <- sqrt(diag(vcov(g0)))
> b1_Wald <- 2*pnorm(abs((coef0[1]-1)/stder0[1]),lower.tail=FALSE)
> b2_Wald <- 2*pnorm(abs((coef0[2]-1)/stder0[2]),lower.tail=FALSE)
```

Do we think this is cheating or not? To get the appropriate confidence intervals to which this coverage would apply, we would need to be able to compute profile confidence intervals. It might in fact be easier, and more useful in the long run, to write the code to compute profile confidence intervals than to understand why the curvature of the likelihood surface is overestimated (and hence the standard errors are underestimated → confidence intervals are too small → coverage is low) in `glmer`.



To do: The double points for Wald/glmer_4.L (solid blue points) occur because I computed the Wald CIs in two different runs, two different ways. I thought that the results should be *identical*, but various changes may have led to different random number seeding/sequences. In general the results seem within binomial error for 500 samples (although $n = 50$, $\tau = 2$ seems a bit extreme).

To do: Compute bias and MSE?

To do: add binomial errors (and position_dodge to coverage plot?)

To do: Re-do all runs with 1000 (rather than 500) MCMC runs — don't expect anything to change, but should do it for consistency.

To do: screen for warnings/errors; count them and exclude from calculations?

Tables from Zhang et al:

- table I: $\beta = (1, 1)$, $n = 50$
- table II: $\beta = (1, 1)$, $n = 100$
- table III: $\beta = (1, 1)$, $n = 500$
- table IV: $\beta = (0, 0)$, $n = 500$
- table V: $\beta = (1.5, 1.5)$, $n = 500$

- table VI: $\beta = (2, 2)$, $n = 500$
- table VII: respiratory data
- table VIII: $\beta = (1, 1)$, $n = 100$, RE=Gamma

References

Davis, C. S. (1991, December). Semi-parametric and non-parametric methods for the analysis of repeated measurements with applications to clinical trials. *Statistics in Medicine* 10(12), 1959–1980.

Zhang, H., N. Lu, C. Feng, S. W. Thurston, Y. Xia, L. Zhu, and X. M. Tu (2011). On fitting generalized linear mixed-effects models for binary responses using different statistical packages. *Statistics in Medicine*.