Methods in Ecology and Evolution

British Ecological Society

Methods in Ecology and Evolution 2016, **7,** 493–498 doi: 10.1111/2041-210X.12504

APPLICATION

SIMR: an R package for power analysis of generalized linear mixed models by simulation

Peter Green* and Catriona J. MacLeod

Landcare Research, Private Bag 1930, Dunedin 9054, New Zealand

Summary

1. The R package SIMR allows users to calculate power for generalized linear mixed models from the LME4 package. The power calculations are based on Monte Carlo simulations.

2. It includes tools for (i) running a power analysis for a given model and design; and (ii) calculating power curves to assess trade-offs between power and sample size.

3. This paper presents a tutorial using a simple example of count data with mixed effects (with structure representative of environmental monitoring data) to guide the user along a gentle learning curve, adding only a few commands or options at a time.

Key-words: experimental design, glmm, Monte Carlo, random effects, sample size, type II error

Introduction

The power of a hypothesis test is defined as the probability that the test will reject the null hypothesis, assuming that the null hypothesis is false. Put another way, if an effect is real, what is the probability that an analysis will judge that the effect is statistically significant?

If a study is underpowered, resources might be wasted and real effects might be missed (Legg & Nagy 2006; Field *et al.* 2007). On the other hand, a large study might be overpowered and so be more expensive than is necessary (Johnson *et al.* 2015). Therefore, it is good practice to perform a power analysis before collecting data, to ensure that the sample has the appropriate size to answer whatever research question is being considered.

Generalized linear mixed models (GLMMs) are important in ecology, allowing the analysis of counts and proportions as well as continuous data (Bolker *et al.* 2009), and controlling for spatial non-independence (Raudenbush & Liu 2000; Rhodes & Jonzén 2011).

Monte Carlo simulation is a flexible and accurate method appropriate for realistic ecological study designs (Bolker 2008; Johnson *et al.* 2015). There are some cases where we could use analytical formulas to calculate the power, but these will usually be an approximation or require a special form for the design (Arnold *et al.* 2011). Simulation is a single method applicable across a wide range of models and methods. Even when formulae are available for a particular model and design, locating and applying the appropriate formula might be difficult enough that simulation is preferred.

For a researcher not sufficiently comfortable in R (R Development Core Team 2015), setting up a simulation experiment could be too complicated (see e.g. Bolker 2008, Chapter 5). Even for someone experienced in R, the time taken setting up the analysis might be better spent elsewhere. In this article we introduce a tool to automate this process.

The SIMR package

There are a range of R packages (see Fig. 1) currently available for power analysis of mixed models (Martin *et al.* 2011; Reich *et al.* 2012; Donohue & Edland 2013; Galecki & Burzykowshi 2013). However, there are none that handle both non-normal response variables and a wide range of fixed and random effect specifications (Johnson *et al.* 2015). SIMR is a power analysis package for R, designed to interoperate with the LME4 package for GLMMs (Bates *et al.* 2015).

SIMR is designed to work with any linear mixed model (LMM) or GLMM that can be fit with either `lmer` or `glmer` from LME4. This allows for a wide range of models with different fixed and random effect specifications. Linear models and generalized linear models using `lm` and `glm` in base R are also supported, to allow for models with no random effects.

There are a range of tests for GLMMs, which can be fast but approximate, or slow but accurate (Bolker *et al.* 2009; Verbeke & Molenberghs 2009). This package has interfaces to a large number of tests for single or multiple fixed or random effects (see Appendix S1, Supporting information), both from LME4 and from external packages (Scheipl, Greven & Kuechenhoff 2008; Halekoh & Højsgaard 2014).

A power analysis in SIMR starts with a model fitted in LME4. This will typically be based on an analysis of data from a pilot study, but more advanced users can create artificial pilot data from scratch (see Appendix S2). This design allows for a gentle learning curve for any users already familiar with LME4.

In SIMR, power is calculated by repeating the following three steps: (i) simulate new values for the response variable using the

*Correspondence author. E-mail: greenp@landcareresearch.co.nz

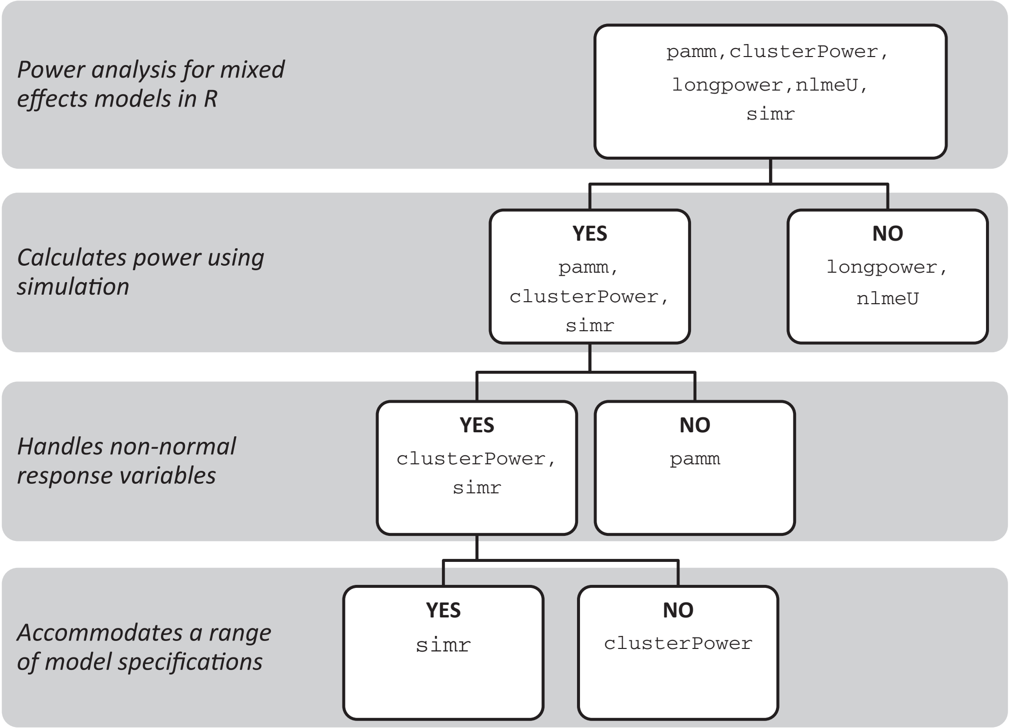
Power analysis for mixed effects models in R
pamm, clusterPower, longpower, nlmeU, simr
Calculates power using simulation
YES
pamm, clusterPower, simr
NO
longpower, nlmeU
Handles non-normal response variables
YES
clusterPower, simr
NO
pamm
Accommodates a range of model specifications
YES
simr
NO
clusterPower

Fig. 1. Assessing the capabilities of R packages for power analysis of mixed effects models: PAMM (Martin 2012), LONGPOWER (Donohue & Edland 2013), CLUSTERPOWER (Reich *et al.* 2012), NLMEU (Galecki & Burzykowshi 2013) and SIMR (this paper).

model provided; (ii) refit the model to the simulated response; (iii) apply a statistical test to the simulated fit. In this setup the tested effect is known to exist, and so every positive test is a true positive and every negative test is a Type II error. The power of the test can be calculated from the number of successes and failures at step 3. More details are given in Appendix S3.

Tutorial

This tutorial illustrates some of the functions available within the SIMR package. Our goal is to provide a gentle learning curve by guiding the user through increasingly complex analyses but adding only a few commands or options at a time.

The tutorial uses the `simdata` data set which is included in the package. The data set is representative of environmental monitoring data, with a response variable z (e.g. bird abundance) measured at 10 levels of the continuous fixed effect variable x (e.g. study year) for three groups g (e.g. study site). There is also a continuous response variable y, which is not used in this tutorial.

Fitting a model

We start by fitting a very simple Poisson mixed effects model in LME4 to the `simdata` data set. In this case we have a random intercept model, where each group (g) has its own intercept but the groups share a common trend.

```
library(simr)
model1 <- glmer(z ~ x + (1|g), family="poisson", data=simdata)
```

```
summary(model1)
## <snip>
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.54079   0.27173   5.670  1.43e-08 ***
## x           -0.11481   0.03955  -2.903    0.0037 **
```

This tutorial focuses on inference about the trend in x. In this case, the estimated effect size for x is $-0{\cdot}11$, which is significant at the 0·01 level using the default z-test.

Note that we have deliberately used a very simple model to make this tutorial easy to follow. A proper analysis would, for example, have a larger number of groups, and would consider problems such as overdispersion. Although a simple model is used for this tutorial, SIMR can handle more complicated specifications (see Appendix S1).

A simple power analysis

Suppose that we wanted to replicate this study. If the effect is real, would we have enough power to expect a positive result?

SPECIFYING AN EFFECT SIZE

Before starting a power analysis, it is important to consider what sort of effect size you are interested in. Power generally increases with effect size, with larger effects being easier to detect. Retrospective 'observed power' calculations, where the target effect size comes from the data, give misleading results (Hoenig & Heisey 2001).

For this example, we will consider the power to detect a slope of −0·05. The fixed effects within the fitted glmer model can be accessed with the LME4 function `fixef`. The SIMR function `fixef<-` can then be used to change the size of the fixed effect. The size of the fixed effect for the variable *x* can be changed from −0·11 to −0·05 as follows:

```
fixef(model1)["x"]
##          x
## -0.1148147
fixef(model1)["x"] <- -0.05
```

In this tutorial, we only change the fixed slope for the variable *x*. However, we could also change the random effect parameters or the residual variance (for models where that is appropriate). See the help entry `?modify` for more details.

RUNNING THE POWER ANALYSIS

Once the model and effect size have been specified, a power analysis is very easy in SIMR. Since these calculations are based on Monte Carlo simulations, your results may be slightly different. If you want to get the same results as the tutorial, you can use `set.seed(123)`.

```
powerSim(model1)
## Power for predictor 'x', (95% confidence interval):
## 33.40% (30.48, 36.42)
##
## Test: z-test
##       Effect size for x is -0.05
##
## Based on 1000 simulations, (5 warnings, 0 errors)[1]
## alpha = 0.05, nrow = 30
##
## Time elapsed: 0 h 3 m 6 s[2]
```

The power to reject the null hypothesis of zero trend in *x* is about 33%, given this particular setup. This would almost always be considered insufficient; traditionally 80% power is considered adequate (although this arbitrary threshold is not always appropriate – see e.g. Field *et al.* 2007).

In practice, the *z*-test might not be suitable for such a small example (Bolker *et al.* 2009). A parametric bootstrap test (e.g. Halekoh & Højsgaard 2014) might be preferred for the final analysis. However, the faster *z*-test is more suitable for learning to use the package and for initial exploratory work during a power analysis. For examples of different test specifications, see Appendix S1 or the help entry `?tests`.

Increasing the sample size

In the first example, estimated power was low. A small pilot study often will not have enough power to detect a small effect, but a larger study might. In SIMR, the `extend` function can be used to add rows to a data frame.

The pilot study had observations at 10 values of *x*, representing for example study years 1 through 10. In this step, we will calculate the effect of increasing this to 20 years.

```
model2 <- extend(model1, along="x", n=20)
powerSim(model2)
## Power for predictor 'x', (95% confidence interval):
## 96.60% (95.28, 97.63)
##
## Test: z-test
##       Effect size for x is -0.05
##
## Based on 1000 simulations, (28 warnings, 0 errors)
## alpha = 0.05, nrow = 60
##
## Time elapsed: 0 h 3 m 37 s
```

The `along` argument specifies which variable is being extended, and `n` specifies how many levels to replace it with. The extended `model2` will now have *x* values from 1 to 20, in three groups as before, for a total of 60 rows (compared to 30 in `model1`).

With observations at 20 values of *x*, we would have plenty of power to detect an effect of size −0·05. In fact, the study might be overpowered with that sample size.

Power analysis at a range of sample sizes

When data collection is costly, the user might want to collect only as much data as are needed to achieve a certain level of statistical power. The `powerCurve` function in SIMR can be used to explore trade-offs between sample size and power.

IDENTIFYING THE MINIMUM SAMPLE SIZE REQUIRED

In the previous example, we found very high power when observations were taken at 20 values of the variable *x*. Could we reduce that number while keeping our power above the usual 80% threshold?

```
pc2 <- powerCurve(model2)
print(pc2)
## Power for predictor 'x', (95% confidence interval),
## by largest value of x:
##      3:  5.70% ( 4.35,  7.32) -  9 rows
##      5:  7.40% ( 5.85,  9.20) - 15 rows
##      7: 15.60% (13.40, 18.00) - 21 rows
##      9: 26.30% (23.59, 29.15) - 27 rows
##     11: 42.70% (39.61, 45.83) - 33 rows
##     12: 52.30% (49.15, 55.44) - 36 rows
##     14: 68.00% (65.01, 70.88) - 42 rows
##     16: 81.60% (79.06, 83.96) - 48 rows
```

[1] Of the 1000 simulations, 5 produced warnings. In this case, the random number generator has thrown up a handful of simulations where the data were a poor fit for the model. Since this only occurred in a very small proportion of cases it is not a cause for concern. To read these warnings, we can use `lastResult()$warnings` (see Appendix S1).

[2] These timings come from a laptop computer with an Intel® Core™ i5-2520M CPU @ 2.50 GHz and 4GB RAM.

```
##     18: 91.30% (89.38, 92.97) - 54 rows
##     20: 96.60% (95.28, 97.63) - 60 rows
##
## Time elapsed: 0 h 20 m 24 s
plot(pc2)
```

Note that we have saved this result to the variable `pc2` to match the numbering in `model2`. Since `model1` did not have sufficient power, we did not run it through `powerCurve`. The plotted output is shown in Fig. 2. We can see that the power to detect a trend in *x* increases with sampling size. The results here were based on fitting the model to 10 different automatically chosen subsets. The smallest subset uses just the first 3 years (i.e. nine observations), and the largest uses the all 20 hypothetical study years (i.e. 60 rows of data). This analysis suggests that the study would have to run for 16 years to have ≥80% power to detect an effect of the specified size.

Varying the number and size of groups

It might not be feasible to increase the number of values of *x* observed. For example, if *x* were study year, we might be unwilling to wait longer for our results. In this case, increasing the number of study sites or the number of measurements at each site might be a better option. These two analyses start back with our original `model1`, which had 10 study years.

ADDING MORE GROUPS

We can add extra levels for *g* the same way we added extra values for *x*. For example if the variable *g* represents our study sites, we could increase the number of sites from 3 to 15.

```
model3 <- extend(model1, along="g", n=15)
pc3 <- powerCurve(model3, along="g")
plot(pc3)
```

The main change from the previous example is that we have passed the variable *g* to the `along` argument. The output for this analysis is shown in Fig. 3. To reach 80% power, we would need at least 11 sites.

INCREASING THE SIZE WITHIN GROUPS

We can replace the `along` argument to `extend` and `powerCurve` with the `within` argument to increase the sam-

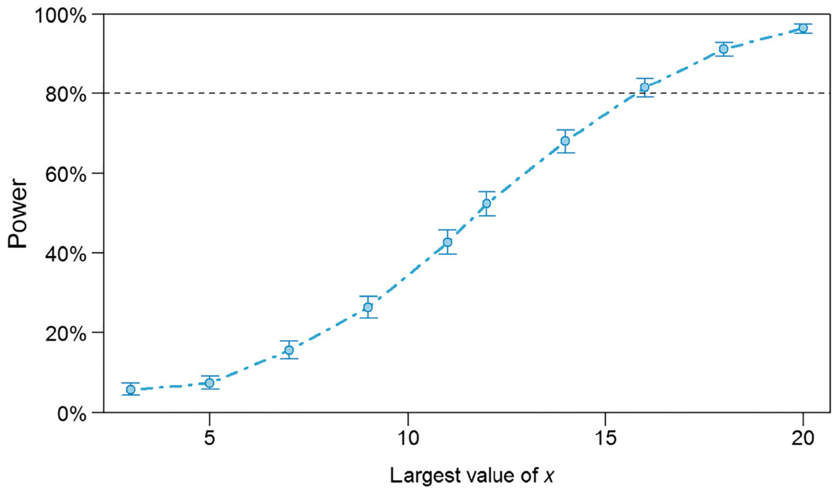

Fig. 2. Power (±95% CI) to detect a fixed effect with size −0·05, calculated over a range of sample sizes using the `powerCurve` function. The number of distinct values for the variable *x* is varied from 3 ($n = 9$) to 20 ($n = 60$).

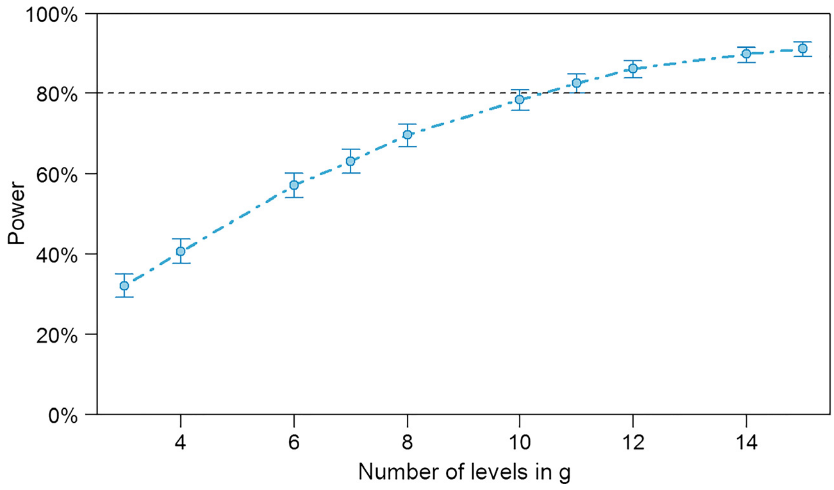

Fig. 3. Power (±95% CI) to detect a fixed effect with size −0·05, calculated over a range of sample sizes using the `powerCurve` function. The number of levels for the factor *g* is varied from 3 ($n = 30$) to 15 ($n = 150$).

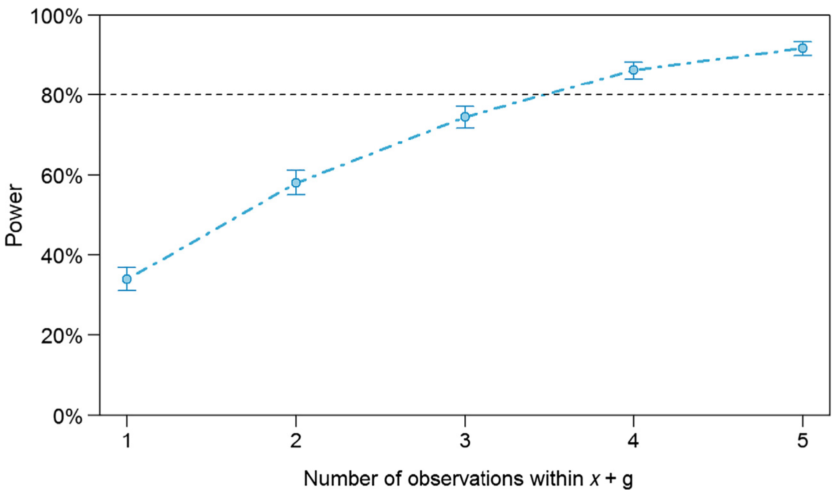

Fig. 4. Power (±95% CI) to detect a fixed effect with size −0·05, calculated over a range of sample sizes using the `powerCurve` function. The number of observations at each combination of *x* and *g* is varied from 1 ($n = 30$) to 5 ($n = 150$).

ple size within groups. Each group has only one observation at each level of *x* and *g*. We can extend this to five observations per site per year as follows:

```
model4 <- extend(model1, within="x+g", n=5)
pc4 <- powerCurve(model4, within="x+g", breaks=1:5)
print(pc4)
## Power for predictor 'x', (95% confidence interval),
## by number of observations within x+g:
##       1: 33.90% (30.97, 36.93) - 30 rows
##       2: 58.00% (54.87, 61.08) - 60 rows
##       3: 74.40% (71.58, 77.08) - 90 rows
##       4: 86.30% (84.01, 88.37) - 120 rows
##       5 : 91.80% (89.92, 93.43) - 150 rows
##
## Time elapsed: 0 h 11 m 35 s
plot(pc4)
```

Note the `breaks` argument to `powerCurve`. This overrides the default behaviour, and gives us one through five observations per combination of *x* and *g*. Figure 4 shows that 4 observations per site per year would give us 80% power.

Other features

The `powerSim` function assumes a number of default settings to make it simple to use, but it can be modified to meet specific needs. For example, users may alter the random number seed for reproducible results (`seed`), or the nominal confidence level (`alpha`), or, by modifying the `nsim` argument from its default setting of 1000, we can increase the precision of our power estimate by increasing the number of simulations. More details can be found in the help with `?powerSim`.

Many of the model parameters can be set using the functions described in the help entry at `?modify`. We modified a fixed effect parameter in the tutorial, but SIMR also has functions for setting random effect variance parameters and residual variances where applicable.

By default, SIMR tests the first fixed effect in a model. However, a wide range of tests can be specified using the `test` argument, including tests for multiple fixed effects and single or multiple random effects. Further examples are provided in the 'Test examples' vignette (Appendix S1), and details of the test functions available in SIMR are available in the help system at `?tests`.

Further work

Version 1.0 of SIMR is designed for any LMM or GLMM fitted using `lmer` or `glmer` in the LME4 package, and for any linear or generalized linear model using `lm` or `glm`, and is focussed on calculating power for hypothesis tests. In future versions we plan to:

- Increase the number of models supported by adding interfaces to additional R packages.
- Extend the package to include precision analysis for confidence intervals.
- Improve the speed of the package by allowing simulations to run in parallel.

Acknowledgements

This research was funded by New Zealand's Ministry of Business, Innovation and Employment (Contract Number AGRB1201). We are grateful to H. Moller, A. Monks, A. Gormley, and D. Tompkins for helpful discussion and to four anonymous reviewers for their detailed and thoughtful feedback.

Data Accessibility

This article uses a randomly generated example data set. The data set, and the code used to create it, is included in the SIMR package, which is available on the CRAN repository at https://cran.r-project.org/web/packages/simr/.

References

Arnold, B.F., Hogan, D.R., Colford, J.M. & Hubbard, A.E. (2011) Simulation methods to estimate design power: an overview for applied research. *BMC Medical Research Methodology*, **11**, 94.

Bates, D., Maechler, M., Bolker, B. & Walker, S. (2015) Fitting linear mixed-effects models using LME4. *Journal of Statistical Software*, **67**, 1–48.

Bolker, B.M. (2008) *Ecological Models and Data in R*. Princeton University Press, Princeton and Oxford.

Bolker, B.M., Brooks, M.E., Clark, C.J., Geange, S.W., Poulsen, J.R., Stevens, M.H.H. & White, J.-S.S. (2009) Generalized linear mixed models: a practical guide for ecology and evolution. *Trends in Ecology and Evolution*, **24**, 127–135.

Donohue, M.C. & Edland, S.D. (2013) LONGPOWER: Power and Sample Size Calculators for Longitudinal Data. R package version 1.0-11. Retrieved from https://cran.r-project.org/web/packages/longpower

Field, S.A., O'Connor, P.J., Tyre, A.J. & Possingham, H.P. (2007) Making monitoring meaningful. *Austral Ecology*, **32**, 485–491.

Galecki, A. & Burzykowshi, T. (2013) *Linear Mixed-Effects Models Using R: A Step-by-Step Approach*. Springer, New York.

Halekoh, U. & Højsgaard, S. (2014) A Kenward-Roger approximation and parametric bootstrap methods for tests in linear mixed models – the R package PBKRTEST. *Journal of Statistical Software*, **59**, 1–30.

Hoenig, J.M. & Heisey, D.M. (2001) The abuse of power: the pervasive fallacy of power calculations for data analysis. *The American Statistician*, **55**, 19–24.

Johnson, P.C.D., Barry, S.J.E., Ferguson, H.M. & Müller, P. (2015) Power analysis for generalized linear mixed models in ecology and evolution. *Methods in Ecology and Evolution*, **6**, 133–142.

Legg, C.J. & Nagy, L. (2006) Why most conservation monitoring is, but need not be, a waste of time. *Journal of Environmental Management*, **78**, 194–199.

Martin, J. (2012) PAMM: power analysis for random effects in mixed models. R package version 0.7. Retrieved from https://cran.r-project.org/web/packages/pamm

Martin, J.G.A., Nussey, D.H., Wilson, A.J. & Réale, D. (2011) Measuring individual differences in reaction norms in field and experimental studies: a power analysis of random regression models. *Methods in Ecology and Evolution*, **2**, 362–374.

R Core Team. (2015) *R: A Language and Environment for Statistical Computing*. R Development Core Team, Vienna, Austria.

Raudenbush, S.W. & Liu, X. (2000) Statistical power and optimal design for multisite randomised trials. *Psychological Methods*, **5**, 199–213.

Reich, N.G., Myers, J.A., Obeng, D., Milstone, A.M. & Perl, T.M. (2012) Empirical power and sample size calculations for cluster-randomized and cluster-randomized crossover studies. *PLoS One*, **7**, e35564.

Rhodes, J.R. & Jonzén, N. (2011) Monitoring temporal trends in spatially structured populations: how should sampling effort be allocated between space and time? *Ecography*, **34**, 1040–1048.

Scheipl, F., Greven, S. & Kuechenhoff, H. (2008) Size and power of tests for a zero random effect variance or polynomial regression in additive and linear mixed models. *Computational Statistics & Data Analysis*, **52**, 3283–3299.

Verbeke, G. & Molenberghs, G. (2009) *Linear Mixed Models for Longitudinal Data*. Springer, New York.

Received 26 August 2015; accepted 25 October 2015
Handling Editor: Shinichi Nakagawa

Supporting Information

Additional Supporting Information may be found in the online version of this article.

Appendix S1 The 'Test examples' vignette, illustrating some of the range of models and tests available in SIMR.

Appendix S2 The 'Power analysis from scratch' vignette, explaining how to start a power analysis without relying on pilot data.

Appendix S3 Additional details about the simulation process and power calculations.