

# Multiple runs of the same parameter combination with R package pse

Chalom, A. <sup>\*†</sup>      Prado, P.I. <sup>\*</sup>

Version 0.3.1, November 23, 2013

This document presents an extension to the practical tutorial found in the vignette “pse\_tutorial” of this package. If you’re not familiar with the basic concepts of parameter space exploration, please refer to the tutorial. To read about the underlying theory, please refer to our work in [1].

The problem we will address here is how to deal with the parameter space exploration of stochastic models. When considering deterministic models, the usual approach to parameter space exploration is to generate a hypercube containing several parameter combinations of interest, running the model with each parameter combination, and summarizing the results with empirical cumulative density functions (ECDFs) and partial rank correlation coefficients (PRCCs) between each model input and output variables. However, if the model isn’t deterministic, a single run of the model may not be able to provide enough information about a particular point in the parameter space. Thus, it is necessary to run the model several times at the same combinations, and summarize the information for each point before proceeding to the uncertainty and sensibility analyses. This approach is discussed in [2], and our terminology is based on this paper. It should be noted that these problems are part of a fresh and open area of study, both in the biology and the engineering communities. The simplest solution is to evaluate average responses from many runs with each combination of parameters. This tutorial presents some simple tools to do this and to evaluate how much of total variation is captured by averaging the responses.

It is usual to refer to the *aleatory* and *epistemic* components of the uncertainty as, respectively, the uncertainty due to random variation of the model behaviour for a fixed combination of parameters and the uncertainty due to our knowledge about the values of the parameters. For comparison, deterministic models only present epistemic uncertainty.

---

<sup>\*</sup>Theoretical Ecology Lab, LAGE at Dep. Ecologia, Instituto de Biociências, Universidade de São Paulo, Rua do Matão travessa 14 nº 321, São Paulo, SP, CEP 05508-900, Brazil.

<sup>†</sup>email: andrechalom@gmail.com

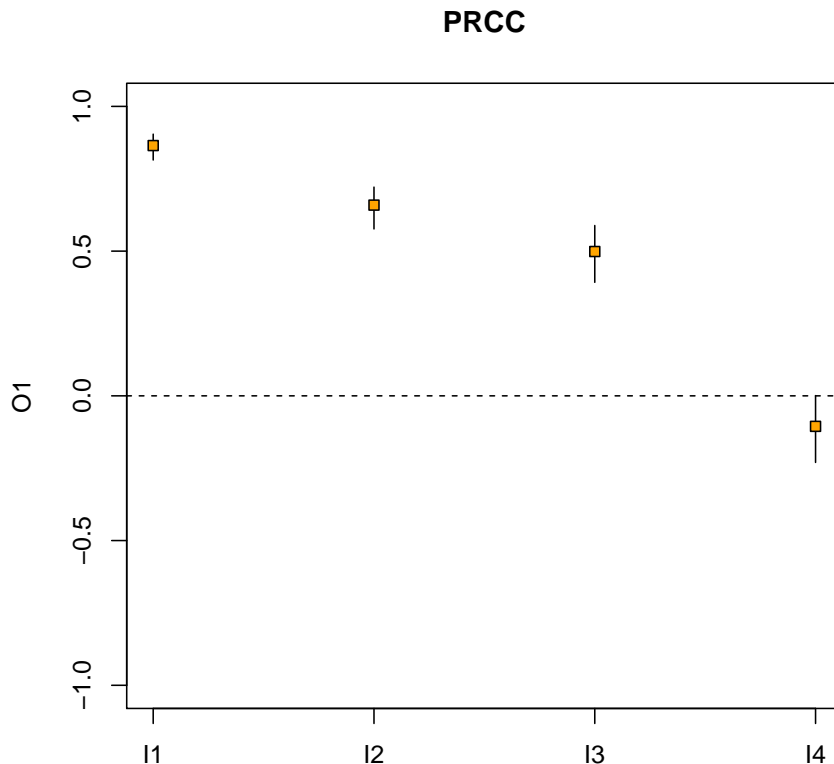
# 1 A simple example

We will show the use of multiple runs in the “pse” package with a very simple model described by:

```
> oneRun <- function (x1, x2, x3, x4)
+   10 * x1 + 5 * x2 + 3 * rnorm(1, x3, x4)
> modelRun <- function (my.data){
+   mapply(oneRun,
+         my.data[,1], my.data[,2], my.data[,3], my.data[,4])
+ }
```

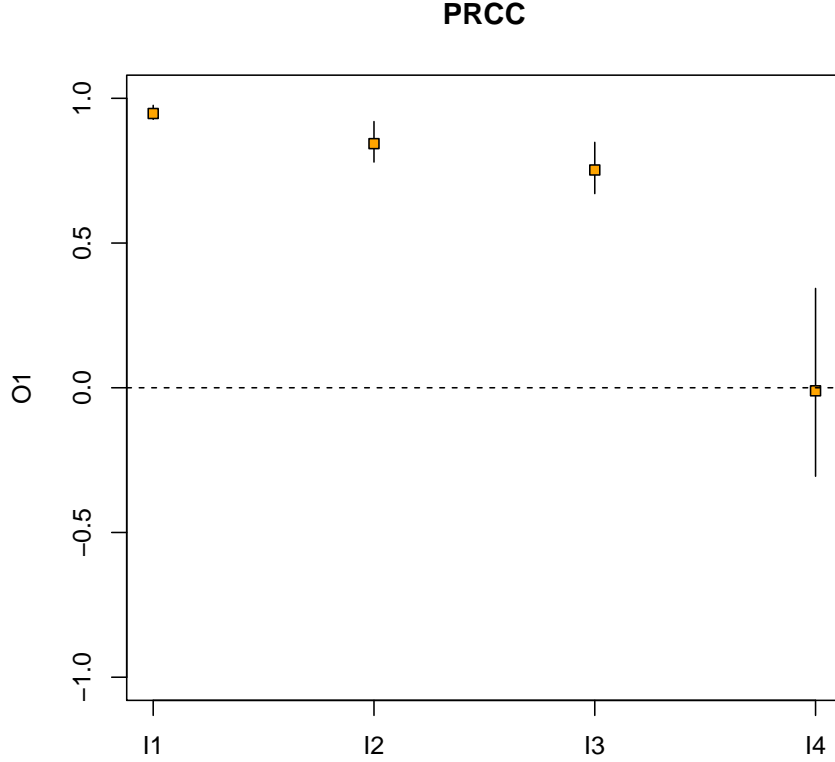
We then generate a Latin Hypercube where parameters  $x_i$  will be varied uniformly between 0 and 1. The code for a hypercube with a single run at each point and resulting partial correlation plot is:

```
> library(pse)
> LHS1 <- LHS(modelRun, N=300, factors=4, nboot=50)
> plotprcc(LHS1)
```



Now, we take a look at the same model, but *repeating* the simulation several times for each data point and averaging the results. The *repetitions* argument sets the number of evaluations for each combination of parameters. Note that the total number of model evaluations ( $N \cdot \text{repetitions}$ ) is the same, in our case, as the LHS1 defined above:

```
> LHS2 <- LHS(modelRun, N=60, factors=4, repetitions=5, nboot=50)
> plotprcc(LHS2)
```



It should be clear from the graphs that the results we get from both schemes (which we will call single-run and repetition schemes) are different, even for a simple model like this. The repetition scheme usually results in larger values of PRCC for variables which are actually correlated to the model output, by mitigating the aleatory uncertainty in each data point. However, this comes at a cost of having less samples to use in statistical inference (resulting in loss of the power of significance tests, or in our case, an increase in the bootstrapped confidence intervals).

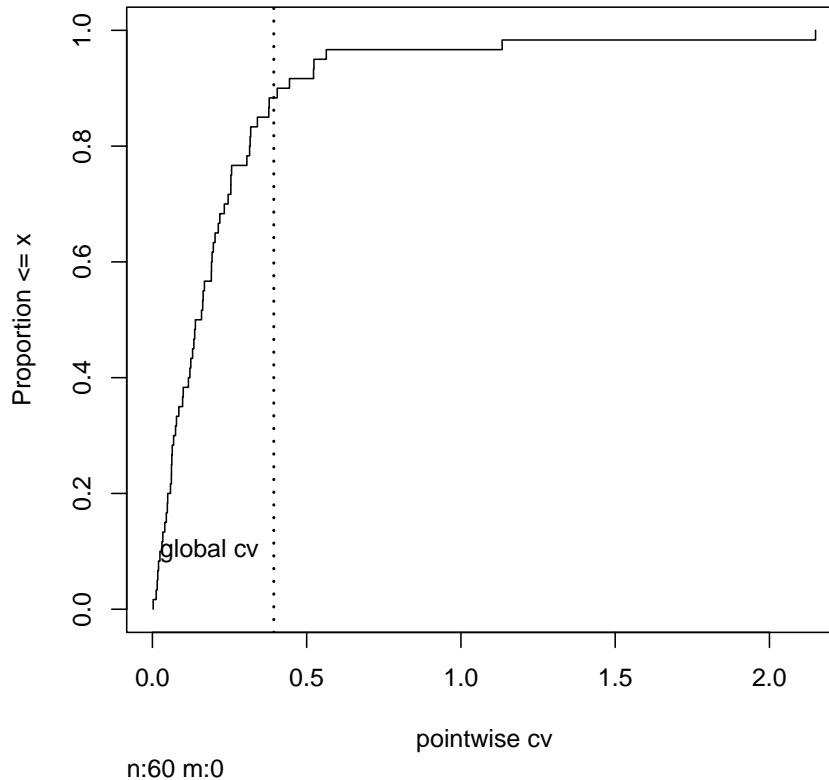
The repetition scheme also has the advantage that it permits a crude estimate of the aleatory uncertainty by means of the *coefficients of variation* (cv). The **cv** and **plotcv** functions can be used to identify whether the aleatory variability is comparable to the total model variability. This plot presents an empirical cumulative distribution function (ecdf) of the variation of the model responses obtained for each parameter combination (pointwise cv). The dotted vertical line corresponds to the variation of the average model response through all combinations (global cv). If the global cv is far greater than all of the pointwise cvs, this means that the epistemic variability is far greater than the aleatory variation for any point. In contrast, if the global cv appears to the left of the graph, thus being smaller than most pointwise cvs, this is probably a sign that the aleatory variation may be masking the effect of the parameter variation, and so the sensitivity analyses will probably be compromised.

In our example, the cv of the whole result set is comparatively larger than most of the pointwise cvs. It is then reasonable to assume that the uncertainty and sensitivity analyses done with average responses are robust.

```
> cv(get.results(LHS2)) # global CV
```

```
[1] 0.3934826
```

```
> plotcv(LHS2)
```



One caveat of using the **cv** is that it is only meaningful if the distribution of results for a given point in the parameter space is unimodal. It is strongly recommended that you check the model behaviour for multimodality before applying any of the uncertainty and sensitivity analyses discussed here. Strong multimodality is often evident from the scatterplots generated for single-run hypercubes.

## 2 Uncoupling analyses

The **tell** method of the LHS package can be used several times to add repetitions to an object. This can be done to uncouple the generation of the LHS and the running of the model, or even to provide more data points to a hypercube. For example, to add a repetition to the LHS2 object, simply execute:

```
> newdata <- modelRun(get.data(LHS2))  
> LHS3 <- tell(LHS2, newdata)
```

This can be used to iteratively add repetitions until the PRCC is stable. In this example, the PRCC scores show very little change after 6 to 7 repetitions. This procedure can be coupled with SMBA evaluations between different hypercube sizes to find an acceptable bound for the number of total model evaluations.

## References

- [1] A. Chalom and P.I.K.L. Prado, *Parameter space exploration of ecological models*, arXiv:1210.6278 [q-bio.QM], 2012.
- [2] S. Marino, I.B. Hogue, C.J. Ray and D.E. Kirschner, *A methodology for performing global uncertainty and sensitivity analysis in systems biology*, Journal of Theoretical Biology 254: 178–196, 2008.