An Introduction to MBHdesign

Scott D. Foster
CSIRO, Hobart, Tasmania, Australia

Abstract

The MBHdesign package is useful for creating spatially balanced designs, especially when legacy sites are present. The package implements the methods described in Foster *et al.* (in review), which is an extension of Balanced Adaptive Sampling (Robertson *et al.* 2013, BAS).[*] In this tutorial, we will go through the three steps of:

1. Altering inclusion probabilities for spatial balance, taking into account the location of legacy sites. This is done using the function `alterInclProbs`;
2. Generating spatially balanced designs for a given set of inclusion probabilities, through the function `quasiSamp`; and
3. Analysing some (made up) data using model-based methods (using `modEsti`).

Keywords: Spatially-Balanced Survey Design, Balanced Adaptive Sampling, Spatially Correlated Poisson Sampling, GRTS, R.

First Things First

Before starting with this introduction to `MBHdesign`, we need to make sure that everything is set up properly. Much of this will vary from computer to computer, but you must have a working version of R installed (preferably the latest one). At the time of writing, the latest version was R-3.3.2. It does not matter whether you prefer to use R through a development environment (such as RStudio) or through the command line – the results will be the same. So, start R and then:

```
install.packages( "MBHdesign")
```

You will be asked which repository you want to use. Just use one that is geographically close to where you are (or where your computer is). Next load the package.

```
library( MBHdesign)
```

For illustration is is also good to fix the random number seed, so that this document is reproducible *exactly*.

[*] although the function `quasiSamp()` is the only function that directly contains the idea in BAS

```
set.seed( 747)  #a 747 is a big plane
```

Now, we are good to go with the rest of the introduction.

The Illustrative Design Scenario

Let's pretend that we want to generate $n = 10$ samples on a grid of points (representing the centres of a tessellation). The grid of points consists of $N = 100 \times 100 = 10000$ points in 2-dimensional space (spanning the interval $[0, 1]$ in both dimensions). Let's also pretend that there are 3 legacy sites, that have been sampled in previous survey efforts, and we wish to revisit them in the current survey. The legacy sites are located at random throughout the study area. Here, I have generated it all in R (painstakingly), but in a real application, most of this information could be read in from file.

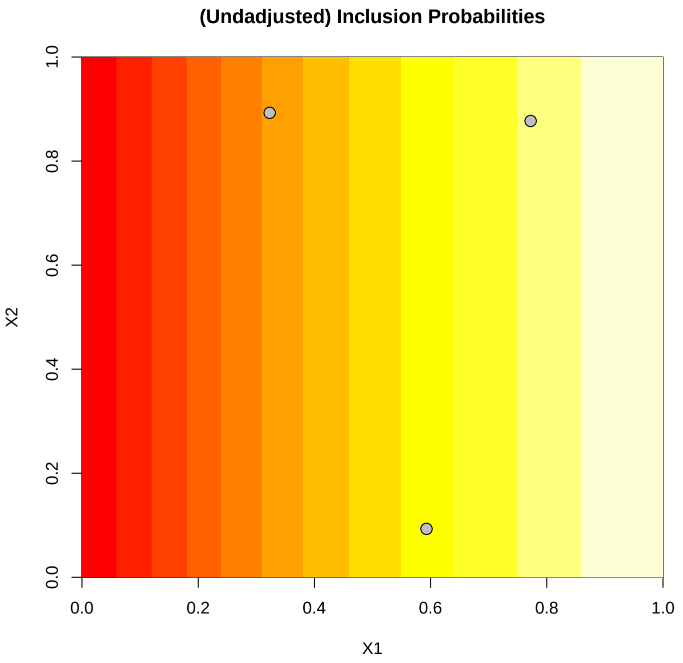
```
#number of samples
n <- 10
#number of points to sample from
N <- 100^2
#the sampling grid (offset so that the edge locations have same area)
offsetX <- 1/(2*sqrt( N))
my.seq <- seq( from=offsetX, to=1-offsetX, length=sqrt(N))
X <- expand.grid( my.seq, my.seq)
#the legacy sites (three of them)
legacySites <- matrix( runif( 6), ncol=2, byrow=TRUE)
#names can be useful
colnames( X) <- colnames( legacySites) <- c("X1","X2")
```

Inclusion Probabilities

Key to this whole design process is the concept of inclusion probabilities. Inclusion probabilities define the chance that any particular site will be part of the sample. So, if a site's inclusion probability is small, then the site is unlikely to be included into the sample. Specifying inclusion probabilities can improve efficiency of the sampling design. That is, standard errors can be reduced for a given number of samples. The 'trick' is to specify inclusion probabilities so that the sites that should have highly variable observations are sampled more often (e.g. Grafström and Tillé 2013). In ecology, variance often increases with abundance (due to Taylor's Power Law; Taylor 1961), so inclusion probabilities could be increased with abundance. If there is no knowledge about the area being sampled, then all sites should be given equal inclusion probabilities (equal to n/N). The only formal requirement, in terms of `MBHdesign`, is that the inclusion probabilities must sum to n.

Here, we are going to pretend that there is some gradient in the variance of the population under study. We stress that this is illustrative only.

```
#non-uniform inclusion probabilities
inclProbs <- 1-exp(-X[,1])
#scaling to enforce summation to n
inclProbs <- n * inclProbs / sum( inclProbs)
#uniform inclusion probabilities would be inclProbs <- rep( n/N, times=N)
#visualise
image( x=unique( X[,1]), y=unique( X[,2]),
    z=matrix( inclProbs, nrow=sqrt(nrow(X)), ncol=sqrt(nrow( X))),
    main="(Undadjusted) Inclusion Probabilities",
    ylab=colnames( X)[2], xlab=colnames( X)[1])
#The legacy locations
points( legacySites, pch=21, bg=grey(0.75), cex=1.5)
```

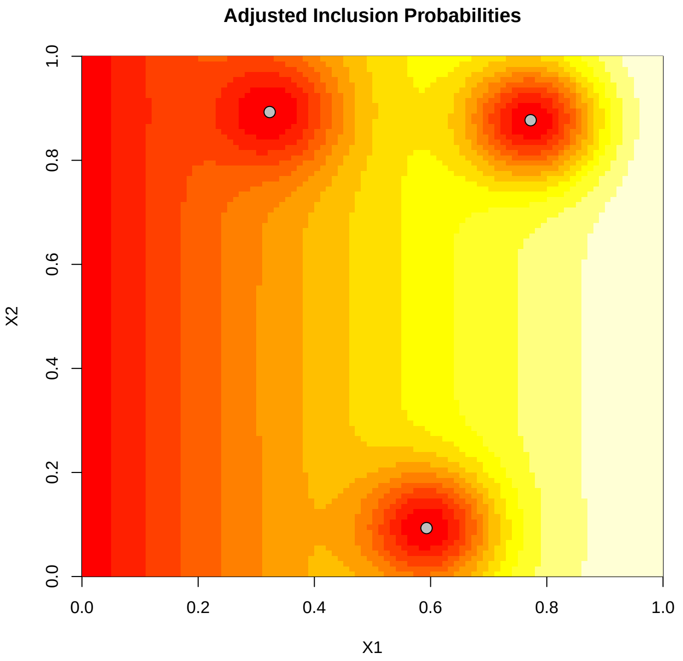


Accommodating Legacy Sites

To generate a design that is spatially balanced *in both the n new sample sites and the legacy sites*, we adjust the inclusion probabilities. The adjustment (see Foster *et al.* in review) reduces the inclusion probabilities so that sites near legacy sites are less likely to be chosen in the new sample.

```
#alter inclusion probabilities
#   so that new samples should be well-spaced from legacy
altInclProbs <- alterInclProbs( legacy.sites=legacySites,
                                potential.sites=X, inclusion.probs = inclProbs)
#visualise
```

```
image( x=unique( X[,1]), y=unique( X[,2]),
    z=matrix( altInclProbs, nrow=sqrt(nrow(X)), ncol=sqrt(nrow( X))),
    main="Adjusted Inclusion Probabilities",
    ylab=colnames( X)[2], xlab=colnames( X)[1])
#The legacy locations
points( legacySites, pch=21, bg=grey(0.75), cex=1.5)
```



So, the inclusion probabilities have been reduced around the legacy sites. It is perhaps worth noting that the reduction in inclusion probabilities, due to the legacy sites, can be viewed as *sequential*. This means that the reduction for any legacy site is in addition to the reduction of all of the other legacy sites – there is no extra joint effect. Also, the adjustment is proportional to the original inclusion probability, so that a small inclusion probability and a large inclusion probability are both adjusted proportionally to the same amount.

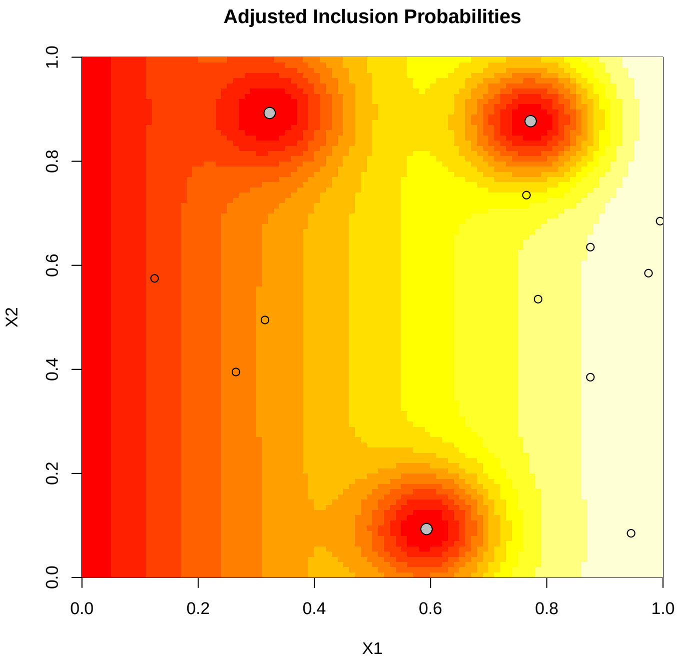
There are some other arguments to the `altInclProbs()` function (omitted for clarity here). These can be seen to refine the call and/or to make the computer to do its work quicker. Type `?altInclProbs` for more details.

Generating the Design

Irrespective of how the inclusion probabilities were obtained, we can now use them to generate a spatially balanced design.

```
#generate the design according to the altered inclusion probabilities.
samp <- quasiSamp( n=n, dimension=2,
        study.area=matrix( c(0,0, 0,1, 1,0, 1,1),ncol=2,  byrow=TRUE),
        potential.sites=X, inclusion.probs=altInclProbs)
```

```
#visualise
image( x=unique( X[,1]), y=unique( X[,2]),
    z=matrix( altInclProbs, nrow=sqrt(nrow(X)), ncol=sqrt(nrow( X))),
    main="Adjusted Inclusion Probabilities",
    ylab=colnames( X)[2], xlab=colnames( X)[1])
#The legacy locations
points( legacySites, pch=21, bg=grey(0.75), cex=1.5)
points( samp[,1:2], pch=21)
```



Voilà! A spatially balanced design that incorporates legacy sites. It is contained in the object `samp`, which looks like:

```
print( samp, row.names=FALSE)

    X1    X2 inclusion.probabilities   ID
 0.875 0.635            0.0017984472 6388
 0.995 0.685            0.0019495731 6900
 0.125 0.575            0.0003638119 5713
 0.875 0.385            0.0018055266 3888
 0.315 0.495            0.0008366335 4932
 0.945 0.085            0.0018926965  895
 0.785 0.535            0.0016838377 5379
 0.975 0.585            0.0019282792 5898
 0.265 0.395            0.0007207841 3927
 0.765 0.735            0.0013289274 7377
```

The columns of `samp` are:

- The sample locations in the `X1` and `X2` dimensions;
- The inclusion probability for that sampling location; and
- The row number (ID), of the original list of potential sites (X).

Analysis

After finalising the design, time comes to go and undertake the survey. For illustration, we do this *in silico* and generate observations according to a pre-defined function (following Foster *et al.* in review, amongst others).

```
#generate some `observations' for the new sites
Z <- 3*( X[samp$ID,1]+X[samp$ID,2]) +
                        sin( 6*( X[samp$ID,1]+X[samp$ID,2]))
#and some for the legacy sites
Zlegacy <- 3*( legacySites[,1]+legacySites[,2]) +
                        sin( 6*( legacySites[,1]+legacySites[,2]))
```

These data can be analysed in two ways: 1) design-based, which uses minimal assumptions about the data; and 2) model-based, which attempts to describe more aspects of the data. See Foster *et al.* (in review) for a more complete description. For design-based analysis we take a weighted average of the estimator for the legacy sites and the estimator for the new sites. In both cases the estimates follow Horvitz and Thompson (1952). Please do read the section in Foster *et al.* (in review) for comments on estimation, it could save you some grief.

```
#the proportion of legacy sites in the whole sample
fracLegacy <- nrow( legacySites) / (n+nrow( legacySites))
#inclusion probabilities for legacy sites
#   (these are just made up, from uniform)
LegInclProbs <- rep( nrow( legacySites) / N, nrow( legacySites))
#estimator based on legacy sites only
legacyHT <- (1/N) * sum( Zlegacy / LegInclProbs)
#estimator based on new sites only
newHT <- (1/N) * sum( Z / samp$inclusion.probabilities)
mean.estimator <- fracLegacy * legacyHT + (1-fracLegacy) * newHT
#print the mean
print( mean.estimator)

[1] 2.718913
```

This is pretty close to the true value of 2.9994. To get a standard error for this estimate, we use the `total.est()` function from the `spsurvey` (Kincaid and Olsen 2015), which implements the neighbourhood estimator of Stevens and Olsen (2003).

```
#load the spsurvey package
library( spsurvey)
#rescale the inclusion probs
#   (the sample frames are the same in legacy and new sites)
tmpInclProbs <- ( c( samp$inclusion.probabilities, LegInclProbs) / n) *
                                                (n+nrow(legacySites))
#calculate the standard error
se.estimator <- total.est( z=c(Z, Zlegacy),
        wgt=1/tmpInclProbs,
        x=c(X[samp$ID,1], legacySites[,1]),
        y=c(X[samp$ID,2], legacySites[,2]))$StdError[2]
#print it
print( se.estimator)

[1] 0.4291556
```

For model-based mean and standard errors we follow the ‘GAMdist’ approach in Foster *et al.* (in review).

```
tmp <- modEsti( y=c( Z, Zlegacy), locations=rbind( X[samp$ID,], legacySites),
        includeLegacyLocation=TRUE, legacyIDs=n + 1:nrow( legacySites),
        predPts=X, control=list(B=1000))
print( tmp)

$mean
[1] 2.734417

$se
[1] 0.2752807

$CI
    2.5%    97.5%
2.170967 3.282242
```

In this case, the standard error estimates are quite different. On average, they tend to be (when there are only a few legacy sites). Even so, this level of difference is unusual.

Last Things Last

The only remaining thing to do is to tidy up our workspace. First, to export our sample locations. Second, to remove all objects for this analysis from your workspace.

```
#write csv
write.csv( samp, file="sample1.csv", row.names=FALSE)
#tidy
rm( list=ls())
```

References

Foster S, Hosack G, Lawrence E, Przeslawski R, Hedge P, Caley M, Barrett N, Williams A, Li J, Lynch T, Dambacher J, Sweatman H, Hayes K (in review). "Spatially-Balanced Designs that Incorporate Legacy Sites."

Grafström A, Tillé Y (2013). "Doubly balanced spatial sampling with spreading and restitution of auxiliary totals." *Environmetrics*, **24**(2), 120–131. ISSN 1099-095X. `doi:10.1002/env.2194`. URL `http://dx.doi.org/10.1002/env.2194`.

Horvitz D, Thompson D (1952). "A Generalization of Sampling Without Replacement From a Finite Universe." *Journal of the American Statistical Association*, **47**(260), 663–685. ISSN 01621459. URL `http://www.jstor.org/stable/2280784`.

Kincaid T, Olsen A (2015). *spsurvey: Spatial Survey Design and Analysis*. R package version 3.1, URL `http://www.epa.gov/nheerl/arm/`.

Robertson BL, Brown JA, McDonald T, Jaksons P (2013). "BAS: Balanced Acceptance Sampling of Natural Resources." *Biometrics*, **69**(3), 776–784. ISSN 1541-0420. `doi:10.1111/biom.12059`. URL `http://dx.doi.org/10.1111/biom.12059`.

Stevens D, Olsen A (2003). "Variance estimation for spatially balanced samples of environmental resources." *Environmetrics*, **14**(6), 593–610. ISSN 1099-095X. `doi:10.1002/env.606`. URL `http://dx.doi.org/10.1002/env.606`.

Taylor L (1961). "Aggregation, Variance and the Mean." *Nature*, **189**(4766), 732–735. URL `http://dx.doi.org/10.1038/189732a0`.

1. Appendix

1.1. Computational details

This vignette was created using the following R and add-on package versions

- R version 3.3.2 (2016-10-31), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_AU.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_AU.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_AU.UTF-8`, `LC_MESSAGES=en_AU.UTF-8`, `LC_PAPER=en_AU.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_AU.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: MBHdesign 1.0.60, knitr 1.14, sp 1.2-3, spsurvey 3.3
- Loaded via a namespace (and not attached): Formula 1.2-1, Hmisc 3.17-4, MASS 7.3-45, Matrix 1.2-7.1, RColorBrewer 1.1-2, Rcpp 0.12.7, acepack 1.3-3.3, chron 2.3-47, class 7.3-14, cluster 2.0.5, codetools 0.2-15, colorspace 1.2-6,

data.table 1.9.6, deldir 0.1-9, digest 0.6.8, evaluate 0.9, foreign 0.8-67, formatR 1.4, geometry 0.3-6, ggplot2 2.1.0, grid 3.3.2, gridExtra 2.2.1, gtable 0.1.2, highr 0.6, lattice 0.20-33, latticeExtra 0.6-26, magic 1.5-6, magrittr 1.5, mgcv 1.8-15, munsell 0.4.2, mvtnorm 1.0-3, nlme 3.1-125, nnet 7.3-12, parallel 3.3.2, plyr 1.8.2, randtoolbox 1.16, rgeos 0.3-11, rngWELL 0.10-3, rpart 4.1-10, scales 0.4.0, splines 3.3.2, stringi 1.1.1, stringr 1.0.0, survival 2.39-5, tools 3.3.2

Affiliation:

Scott D. Foster
CSIRO
Marine Laboratories
GPObox 1538
Hobart 7001
Australia E-mail: scott.foster@csiro.au