Overview of Unmarked:
An R Package for the Analysis of Data from Unmarked Animals

Ian Fiske and Richard Chandler

January 27, 2015

Abstract

Unmarked aims to be a complete environment for the statistical analysis of data from surveys of unmarked animals. Currently, the focus is on hierarchical models that separately model a latent state (or states) and an observation process. This vignette provides a brief overview of the package — for a more thorough treatment see [2]

1 Overview of unmarked

Unmarked provides methods to estimate site occupancy, abundance, and density of animals (or possibly other organisms/objects) that cannot be detected with certainty. Numerous models are available that correspond to specialized survey methods such as temporally replicated surveys, distance sampling, removal sampling, and double observer sampling. These data are often associated with metadata related to the design of the study. For example, in distance sampling, the study design (line- or point-transect), distance class break points, transect lengths, and units of measurement need to be accounted for in the analysis. Unmarked uses S4 classes to store data and metadata in a way that allows for easy data manipulation, summarization, and model specification. Table 1 lists the currently implemented models and their associated fitting functions and data classes.

Model	Fitting Function	Data	Citation
Occupancy	occu	unmarkedFrameOccu	[4]
Royle-Nichols	occuRN	unmarkedFrameOccu	[8]
Point Count	pcount	unmarkedFramePCount	[6]
Distance-sampling	distsamp	unmarkedFrameDS	[7]
Generalized distance-sampling	gdistsamp	unmarkedFrameGDS	[1]
Arbitrary multinomial-Poisson	multinomPois	unmarkedFrameMPois	[5]
Colonization-extinction	colext	unmarkedMultFrame	[3]
Generalized multinomial-mixture	gmultmix	unmarkedFrameGMM	[5]

Table 1: Models handled by unmarked.

Each data class can be created with a call to the constructor function of the same name as described in the examples below.

2 Typical unmarked session

The first step is to import the data into R, which we do below using the **read.csv** function. Next, the data need to be formatted for use with a specific model fitting function. This can be accomplished with a call to the appropriate type of **unmarkedFrame**. For example, to prepare the data for a single-season site-occupancy analysis, the function **unmarkedFrameOccu** is used.

2.1 Importing and formatting data

```
> library(unmarked)
> wt <- read.csv(system.file("csv","widewt.csv", package="unmarked"))
```

```
> y <- wt[,2:4]
> siteCovs <-  wt[,c("elev", "forest", "length")]
> obsCovs <- list(date=wt[,c("date.1", "date.2", "date.3")],
     ivel=wt[,c("ivel.1",  "ivel.2", "ivel.3")])
> wt <- unmarkedFrameOccu(y = y, siteCovs = siteCovs, obsCovs = obsCovs)
> summary(wt)
unmarkedFrame Object

237 sites
Maximum number of observations per site: 3
Mean number of observations per site: 2.81
Sites with at least one detection: 79

Tabulation of y observations:
   0    1 <NA>
 483  182   46

Site-level covariates:
      elev               forest              length
 Min.   :-1.436125   Min.   :-1.265352   Min.   :0.1823
 1st Qu.:-0.940726   1st Qu.:-0.974355   1st Qu.:1.4351
 Median :-0.166666   Median :-0.064987   Median :1.6094
 Mean   : 0.007612   Mean   : 0.000088   Mean   :1.5924
 3rd Qu.: 0.994425   3rd Qu.: 0.808005   3rd Qu.:1.7750
 Max.   : 2.434177   Max.   : 2.299367   Max.   :2.2407

Observation-level covariates:
      date               ivel
 Min.   :-2.90434   Min.   :-1.7533
 1st Qu.:-1.11862   1st Qu.:-0.6660
 Median :-0.11862   Median :-0.1395
 Mean   :-0.00022   Mean   : 0.0000
 3rd Qu.: 1.30995   3rd Qu.: 0.5493
 Max.   : 3.80995   Max.   : 5.9795
 NA's   :42         NA's   :46
```

Alternatively, the convenience function **csvToUMF** can be used

```
> wt <- csvToUMF(system.file("csv","widewt.csv", package="unmarked"),
                 long = FALSE, type = "unmarkedFrameOccu")
```

If not all sites have the same numbers of observations, then manual importation of data in long format can be tricky. **csvToUMF** seamlessly handles this situation.

```
> pcru <- csvToUMF(system.file("csv","frog2001pcru.csv", package="unmarked"),
                   long = TRUE, type = "unmarkedFrameOccu")
```

To help stabilize the numerical optimization algorithm, we recommend standardizing the covariates.

```
> obsCovs(pcru) <- scale(obsCovs(pcru))
```

2.2 Fitting models

Occupancy models can then be fit with the occu() function:

```
> fm1 <- occu(~1 ~1, pcru)
> fm2 <- occu(~ MinAfterSunset + Temperature ~ 1, pcru)
> fm2
Call:
occu(formula = ~MinAfterSunset + Temperature ~ 1, data = pcru)

Occupancy:
```

```
 Estimate    SE    z  P(>|z|)
     1.54 0.292 5.26 1.42e-07

Detection:
               Estimate    SE      z  P(>|z|)
(Intercept)      0.2098 0.206  1.017 3.09e-01
MinAfterSunset  -0.0855 0.160 -0.536 5.92e-01
Temperature     -1.8936 0.291 -6.508 7.60e-11

AIC: 356.7591
```

Here, we have specified that the detection process is modeled with the MinAfterSunset and Temperature covariates. No covariates are specified for occupancy here. See ?occu for more details.

2.3 Back-transforming parameter estimates

Unmarked fitting functions return unmarkedFit objects which can be queried to investigate the model fit. Variables can be back-transformed to the unconstrained scale using backTransform. Standard errors are computed using the delta method.

```
> backTransform(fm2, 'state')
Backtransformed linear combination(s) of Occupancy estimate(s)

 Estimate     SE LinComb (Intercept)
    0.823 0.0425    1.54           1

Transformation: logistic
```

The expected probability that a site was occupied is 0.823. This estimate applies to the hypothetical population of all possible sites, not the sites found in our sample. For a good discussion of population-level vs finite-sample inference, see Royle and Dorazio [9] page 117. Note also that finite-sample quantities can be computed in `unmarked` using empirical Bayes methods as demonstrated at the end of this document.

Back-transforming the estimate of ψ was easy because there were no covariates. Because the detection component was modeled with covariates, p is a function, not just a scalar quantity, and so we need to be provide values of our covariates to obtain an estimate of p. Here, we request the probability of detection given a site is occupied and all covariates are set to 0.

```
> backTransform(linearComb(fm2, coefficients = c(1,0,0), type = 'det'))
Backtransformed linear combination(s) of Detection estimate(s)

 Estimate    SE LinComb (Intercept) MinAfterSunset Temperature
    0.552 0.051    0.21           1              0           0

Transformation: logistic
```

Thus, we can say that the expected probability of detection was 0.552 when time of day and temperature are fixed at their mean value. A predict method also exists, which can be used to obtain estimates of parameters at specific covariate values.

```
> newData <- data.frame(MinAfterSunset = 0, Temperature = -2:2)
> round(predict(fm2, type = 'det', newdata = newData, appendData=TRUE), 2)
  Predicted   SE lower upper MinAfterSunset Temperature
1      0.98 0.01  0.93  1.00              0          -2
2      0.89 0.04  0.78  0.95              0          -1
3      0.55 0.05  0.45  0.65              0           0
4      0.16 0.03  0.10  0.23              0           1
5      0.03 0.01  0.01  0.07              0           2
```

Confidence intervals are requested with confint, using either the asymptotic normal approximation or profiling.

```
> confint(fm2, type='det')
```

```
                      0.025      0.975
p(Int)            -0.1946871  0.6142292
p(MinAfterSunset) -0.3985642  0.2274722
p(Temperature)    -2.4638797 -1.3233511
> confint(fm2, type='det', method = "profile")
Profiling parameter 1 of 3 ... done.
Profiling parameter 2 of 3 ... done.
Profiling parameter 3 of 3 ... done.
                      0.025      0.975
p(Int)            -0.1929210  0.6208837
p(MinAfterSunset) -0.4044794  0.2244221
p(Temperature)    -2.5189984 -1.3789261
```

2.4 Model selection and model fit

Model selection and multi-model inference can be implemented after organizing models using the fitList function.

```
> fms <- fitList('psi(.)p(.)' = fm1, 'psi(.)p(Time+Temp)' = fm2)
> modSel(fms)
                   nPars    AIC  delta   AICwt cumltvWt
psi(.)p(Time+Temp)     4 356.76   0.00 1.0e+00     1.00
psi(.)p(.)             2 461.00 104.25 2.3e-23     1.00
> predict(fms, type='det', newdata = newData)
   Predicted         SE     lower      upper
1 0.98196076 0.01266193 0.9306044 0.99549474
2 0.89123189 0.04248804 0.7763166 0.95084836
3 0.55225129 0.05102660 0.4514814 0.64890493
4 0.15658708 0.03298276 0.1021713 0.23248007
5 0.02718682 0.01326263 0.0103505 0.06948653
```

The parametric bootstrap can be used to check the adequacy of model fit. Here we use a χ^2 statistic appropriate for binary data.

```
> chisq <- function(fm) {
     umf <- getData(fm)
     y <- getY(umf)
     y[y>1] <- 1
     sr <- fm@sitesRemoved
     if(length(sr)>0)
         y <- y[-sr,,drop=FALSE]
     fv <- fitted(fm, na.rm=TRUE)
     y[is.na(fv)] <- NA
     sum((y-fv)^2/(fv*(1-fv)), na.rm=TRUE)
     }
> (pb <- parboot(fm2, statistic=chisq, nsim=100))
Call: parboot(object = fm2, statistic = chisq, nsim = 100)

Parametric Bootstrap Statistics:
   t0 mean(t0 - t_B) StdDev(t0 - t_B) Pr(t_B > t0)
1 356           20.2             15.6       0.0792

t_B quantiles:
     0% 2.5% 25% 50% 75% 97.5% 100%
t*1 299  306 326 334 346   371  385

t0 = Original statistic compuated from data
t_B = Vector of bootstrap samples
```

We fail to reject the null hypothesis, and conclude that the model fit is adequate.

2.5 Derived parameters and empirical Bayes methods

The `parboot` function can be also be used to compute confidence intervals for estimates of derived parameters, such as the proportion of sites occupied PAO $= \sum_i z_i$ where z_i is the true occurrence state at site i, which is unknown at sites where no individuals were detected. The "colext" vignette shows examples of using `parboot` to obtain confidence intervals for such derived quantities. An alternative way achieving this goal is to use empirical Bayes methods, which were introduced in `unmarked` version 0.9-5. These methods estimate the posterior distribution of the latent variable given the data and the estimates of the fixed effects (the MLEs). The mean or the mode of the estimated posterior distibution is referred to as the empirical best unbiased predictor (EBUP), which in `unmarked` can be obtained by applying the `bup` function to the estimates of the posterior distributions returned by the `ranef` function. The following code returns the estimate of PAO and a 90% confidence interval.

```
> re <- ranef(fm2)
> EBUP <- bup(re, stat="mode")
> CI <- confint(re, level=0.9)
> rbind(PAO = c(Estimate = sum(EBUP), colSums(CI)) / 130)
     Estimate        5%       95%
PAO 0.8076923 0.7384615 0.9923077
```

Note that this is similar, but slightly lower than the population-level estimate of ψ obtained above.

A plot method also exists for objects returned by `ranef`, but distributions of binary variables are not so pretty. Try it out on a fitted abundance model instead.

References

[1] R. B. Chandler, J. A. Royle, and D. I. King. Inference about density and temporary emigration in unmarked populations. *Ecology*, 92(7):1429–1435, July 2011.

[2] Ian Fiske and Richard Chandler. **unmarked**: An **R** package for fitting hierarchical models of wildlife occurrence and abundance. *Journal of Statistical Software*, 43(10):1–23, 2011.

[3] Darryl I. MacKenzie, James D. Nichols, James E. Hines, Melinda G. Knutson, and Alan B. Franklin. Estimating site occupancy, colonization, and local extinction when a species is detected imperfectly. *Ecology*, 84(8):2200–2207, 2003.

[4] Darryl I. MacKenzie, James D. Nichols, G. B. Lachman, S. Droege, J. A. Royle, and C. A. Langtimm. Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, 83(8):2248–2255, 2002.

[5] J. A. Royle. Generalized estimators of avian abundance from count survey data. *Animal Biodiversity and Conservation*, 27(1):375–386, 2004.

[6] J. A Royle. N-mixture models for estimating population size from spatially replicated counts. *Biometrics*, 60(1):108–115, 2004.

[7] J. A. Royle, D. K. Dawson, and S. Bates. Modeling abundance effects in distance sampling. *Ecology*, 85(6):1591–1597, 2004.

[8] J. A. Royle and J. D. Nichols. Estimating abundance from repeated presence-absence data or point counts. *Ecology*, 84(3):777–790, 2003.

[9] J.A. Royle and R.M. Dorazio. *Hierarchical modeling and inference in ecology: the analysis of data from populations, metapopulations and communities*. Academic Press, 2008.