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

Ian Fiske and Richard Chandler

December 20, 2011

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.

```
> 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.   :-1.265e+00   Min.   :0.1823
 1st Qu.:-0.940726   1st Qu.:-9.744e-01   1st Qu.:1.4351
 Median :-0.166666   Median :-6.499e-02   Median :1.6094
 Mean   : 0.007612   Mean   : 8.798e-05   Mean   :1.5924
 3rd Qu.: 0.994425   3rd Qu.: 8.080e-01   3rd Qu.:1.7750
 Max.   : 2.434177   Max.   : 2.299e+00   Max.   :2.2407

Observation-level covariates:
      date                 ivel
 Min.   :-2.9043386   Min.   :-1.753e+00
 1st Qu.:-1.1186243   1st Qu.:-6.660e-01
 Median :-0.1186243   Median :-1.395e-01
 Mean   :-0.0002173   Mean   :-3.008e-11
 3rd Qu.: 1.3099471   3rd Qu.: 5.493e-01
 Max.   : 3.8099471   Max.   : 5.980e+00
 NA's   :42.0000000   NA's   : 4.600e+01
```

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

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

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 = "unm
```

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

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

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.

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
```

Because the detection component was modeled with covariates, covariate coefficients must be specified to back-transform. 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
```

A predict method also exists.

```
> 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)            -0.1946872  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
```

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 0.957143378 1.00677814
2 0.89123189 0.04248804 0.807955332 0.97450844
3 0.55225129 0.05102660 0.452239161 0.65226342
4 0.15658708 0.03298276 0.091940874 0.22123328
5 0.02718682 0.01326263 0.001192059 0.05318158
```

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.

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.