Emerging Technologies

"Emerging Technologies" will replace Technological Tools

This new column, to be jointly edited by David Inouye and Sam Scheiner, is aimed at highlighting new or emerging areas of technology and methodology in ecology. Topics may range from hardware to software to statistical analyses, or to technologies that are or could be used in ecology. Some of these will be bleeding-edge developments, but they can include long-standing methods from other fields that have not yet caught on in ecology. Here is your chance to share your little-known favorite method or to show off the secret geek side of your personality.

Articles should be no longer than a few thousand words. A suggested format for such an article is: (1) a brief depiction of the concepts or ideas addressed by the technology or methodology, (2) a description of that technology or methodology, and (3) references, readings, and commercial or noncommercial sources, perhaps with a few sentences about each.

Ideas for articles should be directed to David Inouye (301-405-6946; E-mail: inouye@umd.edu) and Sam Scheiner (703-292-7175; E-mail: sscheine@nsf.gov).

Improving the Presentation of Results of Logistic Regression with R

Introduction

In a recent issue of the *ESA Bulletin*, Smart et al. (2004) proposed an interesting new means of presenting the results of logistic regression, incorporating frequency histograms for each category of the dependent variable and an associated scale on the right-hand axis of the traditional probability plot. The new method of presentation clearly increases the information of the graph, but as they recognize, the manual production of these figures is time consuming. They suggest that software manufacturers should incorporate this type of combination graph in future updates of statistical packages.

In this note I show that we do not have to wait for software updates because we already have an easy means to produce and improve this kind of graph. I also provide some R functions to produce some variants of the combination graph.

An easy R approach

R is a free, open-source environment for statistical computing and graphics (R Development Core Team 2003). Its potential use for ecologists has only been described briefly (Elner 2001, Kangas 2004). Some of the developers of R were also innovators in statistical graphics (e.g., Chambers et al. 1983), so it is not surprising that R has strong capabilities to implement any kind of graphics. But, like the standard statistics packages, R does not have (or at least, I did not find it in the extensive help documentation) a combination graph for logistic regression. However, it has facilities to produce scatterplots and to produce histograms. The difference from other statistics packages (apart from the fact that R is not a "package" but a system or language) is that we can easily access and manipulate the elements of the scatterplots and the histograms and can combine them in a single graph. R also provides the user with a set of functions (e.g., plot, points, lines, axis, polygon, etc.) to modify built-in graphics or to build them from scratch.

In the case of logistic regression the data would usually have two variables: the dependent variable (e.g., coded 0 and 1) and the observed data for the predictor variable (independent variable). The process to build a combination graph in R could be the following:

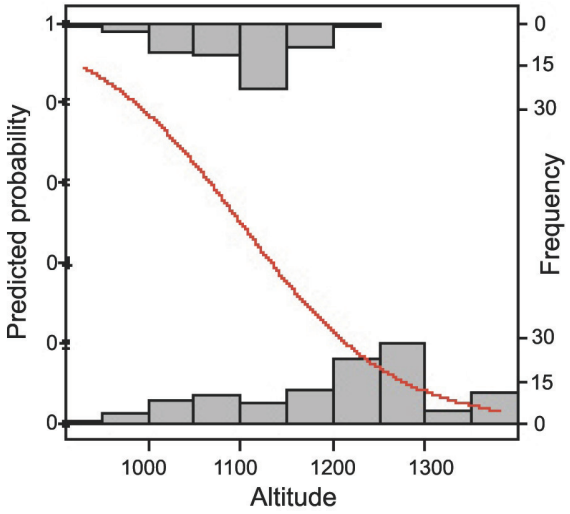
1) Set the draw area with function plot.
2) Use function hist to obtain the boundaries and the counts (i.e., the "heights") of the bins of histograms of the independent variable.
3) Scale the counts to adjust the height of the histograms to the desired height among the 0– 1 scale of the scatterplot. As one of the histograms will be drawn in the top of the graph, subtract from 1 their scaled counts.
4) Use repeatedly the function polygon with the scaled counts and boundaries data to draw the bins of each histogram.
5) Use the function axis and the scaled counts to

draw the right-hand frequency axis.

6) Fit a binomial glm model to the data and add the predicted logistic curve to the graph.

These steps produce the graph of Fig.1, using hypothetical data that describe the probability of occurrence of a tree along an altitudinal gradient.

Fig.1. Fitted logistic regression curve and histograms of both categories of dependent variable.



Some improvements

Although we know now how to produce the combination graph, it is worth remembering that histograms are not the best method for visual description of univariate data. Ellison (1993) gives some reasons to prefer presentations other than histograms. For example, the number of bins in a histogram is something arbitrary (in the above example it was the default of function hist). Summary statistics cannot be computed from the data illustrated in the histogram, and because of the arbitrariness of the bins, the distribution of data is to some degree distorted or exaggerated. Also, histograms hide the raw data, and although we can present a frequency scale, with the reduced graphics of scientific papers it is almost impossible to ascertain the exact number of counts in each bin.

A possible solution to this problem could be to annotate the number of counts in each bin, although it would not solve the problem of the arbitrary bins. From a biological point of view it would sometimes be desirable to summarize the counts in intervals of ecological interest. In R we can both select between a set of algorithms to construct the histogram, and specify the exact sequence of intervals (even of different amplitude). Fig. 2 shows the histograms built for a sequence of intervals of 20 m of altitude.

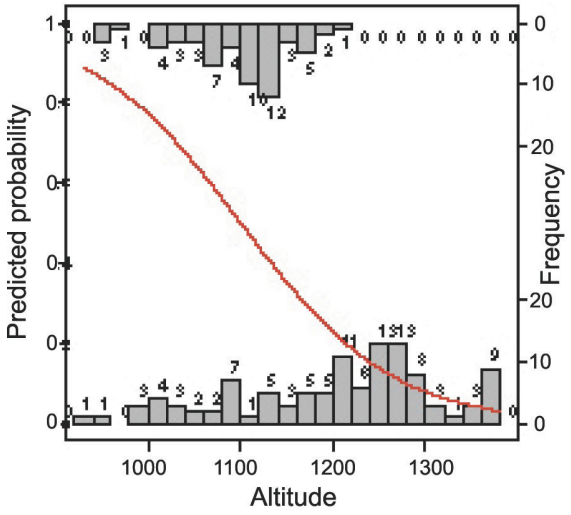


Fig. 2. Fitted logistic regression curve and histograms with bins every 20 m and counts in each bin.

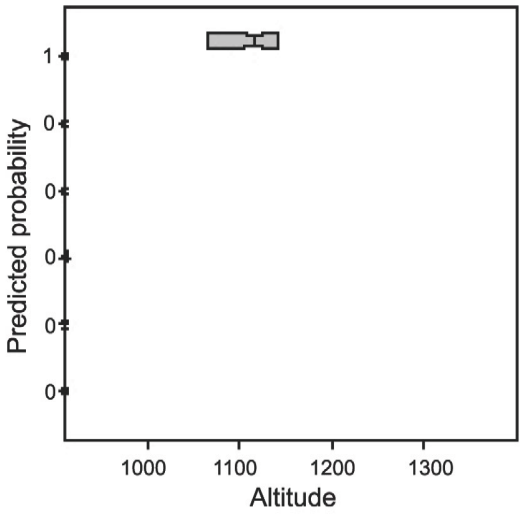


Fig. 3. Fitted logistic Gaussian regression curve with dit plots and box plots of dependent variable categories.

Ellison (1993) suggests the box-and-whisker plot (also called box plot) as an alternative to histograms. Box plots summarize efficiently the information of the data (median, quartiles, ranges, and outliers) and can even present confidence intervals (notched box plots) so that we can compare the distribution of both dependent variable categories. In R, box plots (notched or not) can easily be added to graphs with function boxplot. Fig. 3 shows the possible use of box plots in a combination graph for logistic regression. Another alternative proposed by Ellison is the dit plot. In dit plots each observation is represented by a point placed along the horizontal scale at the exact location of its value. If there are several observations with the same value, they are stacked up (or down) the *y* axis.

In R we can combine dit plots with logistic regression curves following the next steps:

1) Get the unique values with function unique.
2) Get the number of repeated observations for each value with functions unique and length. Add (or subtract in the case of the upper dit plot) a sequential increment to the *y* value of each repeated observation.
3) Represent each observation with function points.
4) Fit a binomial glm model to the data and add the predicted logistic curve to the graph.

With appropriate dit plots we can present the raw data in full; it seems a good alternative (with or without box plots) to histograms in the combined graphs.

It could be even easier than that

Function plot.logi.hist, (Appendix A) is an R function (actually a set of functions) for the naive R user that can be used to produce all the combination graphs mentioned in the text. To produce a combination graph you need only have a working R environment (download it from your nearest mirror site at cran.r-project.org), type or read in your data (you can read your data in several formats, e.g., from a csv or tab-delimited ascii file with read.table; from SAS or SPSS files with library foreign, or from Excel files with library gregmisc), and paste and use function plot.logi.hist. For example, if "tree" is the dependent variable with the presence/absence data and "altitude" the predictor variable with the observational data, typing

```
plot.logi.hist (altitude, tree)
```

will produce a combined graph with logistic curve, dit, and box plots. Other plots and combinations can be produced, adding parameters to the function. For example

```
plot.logi.hist (altitude, tree, type = "hist",
    count.hist = TRUE)
```

will produce the graph with box plot, histograms, and will annotate the counts in each bin.

Graphs can be copied to the clipboard as bitmaps or metafiles or can be saved in a variety of formats, so they can easily be used for papers, presentations, etc.

Like most R functions, plot.logi.hist is a text file; it can be edited with a word processor and customized to accomplish more specific needs of the user.

Literature cited

Chambers, J. M., W. S. Cleveland, B. Kleiner, and P. A. Tukey. 1983. Graphical methods for data analysis. Wadsworth, Belmont, California, USA.

Ellison, A. M. 1993. Exploratory data analysis and graphic display. Pages 14–45 *in* S. M. Scheiner and J. Gurevitch, editors. Design and analysis of ecological experiments. Chapman and Hall, New York, New York, USA.

Ellner, S. P. Review of R, Version 1.1.1. ESA Bulletin **82**:127–128.

Kangas, M. 2004. R: a computational and graphics resource for ecologists. Frontiers in Ecology and the Environment **5**:277.

Smart, J., W. J. Sutherland, A. R. Watkinson, and J. A. Gill. 2004. A new means of presenting the results of logistic regression. ESA Bulletin **85**:100–102.

R Development Core Team. 2003. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ‹http://www.R-project.org›

Marcelino de la Cruz Rot
Departamento de Biologia Vegetal
E.T.S. Ingenieros Agronomos
Universidad Politecnica de Madrid
28040 Madrid Spain
E-mail: marcelino.delacruz@upm.es

Appendices follow...

Appendix A

```
# Function plot.logi.hist  is a set of R functions
# to plot combined graphs for logistic regression. Its
# arguments are: independ (explanatory variable), depend
# (dependent variable), logi.mod (type of fitting, 1 =
# logistic; 2 = "gaussian" logistic), type (type of
# representation, "dit" = dit plot; "hist" = histogram),
# boxp (TRUE = with box plots, FALSE = without), rug
# (TRUE = with rug plots, FALSE = without), las.h
# (orientation of axes labels (0 = vertical, 1 =
# horizontal).



plot.logi.hist <- function (independ, depend, logi.mod = 1,
    type = "dit", boxp = TRUE, rug = FALSE,
    las.h = 1, ...){

# get the label for the x-axis
xlabel <- paste(deparse(substitute(independ)))

# define functions:

# set the draw area if no box plots are to be drawn
 logi.scater <- function (independ, depend, scater = "n",
    x.lab = xlabel, las =las.h){
 plot(independ, depend, cex = 1, type = scater,
    ylab = "Predicted probability", xlab = x.lab,
    cex.lab = 1.5, las = las)
 }

# add rug plot if desired; you could change pch.rug
# (symbol type) or cex.rug (symbol size)
 logi.rug <- function (independ, depend, pch.rug = 16,
    cex.rug = 1){
points(independ, depend, pch = pch.rug ,cex = cex.rug)
 }

# set the draw area and add box plots; you could change
# cold.box (color of the boxes)
logi.box <- function(independ, depend, col.box = "gray",
  x.lab = xlabel, las = las.h){
plot(independ, depend, cex = 1, type = "n",
```

```
   ylim = c(-0.1,1.1), ylab = "Predicted probability",
   xlab = x.lab, cex.lab = 1.5, las = las)
 indep.1 <- independ[depend == 1]
 indep.0 <- independ[depend == 0]
 boxplot(indep.1, horizontal = TRUE, add = TRUE,
    at = 1.05, boxwex = 0.1, col = col.box, notch = T)
boxplot(indep.0, horizontal = TRUE, add = TRUE,
  at = -0.05, boxwex = 0.1, col = col.box, notch = T)
 }

# fit binomial glm and add predicted curve; you could
# change col.cur (color of the curve) or lwd.cur(width
# of the curve)
logi.curve <- function(independ, depend, mod = logi.mod,
    col.cur = "red", lwd.cur = 4){
if (mod == 1) mod3 <- glm(depend ~ independ,
    family = binomial)
if (mod == 2) mod3 <- glm(depend ~ independ +
    I(independ^2), family = binomial)
x.new <- seq(min(independ), max(independ), len = 100)
y.new <- predict(mod3, data.frame(independ = x.new),
    type = "response")
lines(x.new, y.new, lwd = lwd.cur, col = col.cur)
}

# add dit plot; you may want to change pch.dit (type of
# points), cex.p (size of points), and incre (space
# between points)
logi.dit <- function (independ, depend, cex.p = 1,
    pch.dit = 1, incre = 0.02){

 indep.0 <- independ[depend == 0]
 indep.1 <- independ[depend == 1]
 uni.plot.0 <- function(x) length(which(indep.0 == x))
 uni.plot.1 <- function(x) length(which(indep.1 == x))

 # get the number of repeated values of "independ":

 cosa.0 <- apply(as.matrix(unique(indep.0)), 1, uni.plot.0)
 cosa.1 <- apply(as.matrix(unique(indep.1)), 1, uni.plot.1)

 # start ploting:
 points(independ, depend, pch = pch.dit, cex = cex.p)
```

```
 for (i in 1:max(cosa.0)){
     for (j in 1:i){
        points(unique(indep.0)[which(cosa.0 == i+1)],
         rep(0 + incre*j, length(which(cosa.0 == i+1))),
         pch = pch.dit, cex = cex.p)
     }
 }

 for (i in 1:max(cosa.1)){
    for (j in 1:i){
        points(unique(indep.1)[which(cosa.1 == i+1)],
         rep(1 - incre*j, length(which(cosa.1 == i+1))),
         pch = pch.dit, cex = cex.p)
    }
 }
}

# add histograms and frequency axes; you may want to change
# scale.hist (factor to scale histogram height to 0-1
# interval) or col.hist (color of histogram)
 logi.hist <- function(independ, depend, scale.hist = 5,
    col.hist = gray(0.7), count.hist = FALSE,
    intervalo = 0, las.h1 = las.h){

  # get the position of bins
   h.br <- hist(independ, plot = F)$br
   if (intervalo > 0) h.br <- seq(from = range(h.br)[1],
     to = range(h.br)[2], by = intervalo)
   h.x <- hist(independ[depend == 0], breaks = h.br,
     plot = F)$mid

  # get counts in each bin
    h.y0 <- hist(independ[depend == 0], breaks = h.br,
      plot = F)$counts
    h.y1 <- hist(independ[depend == 1], breaks = h.br,
      plot = F)$counts

  # scale the histogram bars to max desired length:
    h.y0n <- h.y0/(max(c(h.y0,h.y1))* scale.hist)
    h.y1n <- 1 - h.y1/(max(c(h.y0,h.y1))* scale.hist)

  # draw bottom histogram:
   for (i in 1:length(h.y0n)){
     if (h.y0n[i] > 0)
```

```
      polygon(c(rep(h.br[i], 2), rep(h.br[i+1], 2)),
        c(0, rep(h.y0n[i], 2), 0), col = col.hist)
   }

  # draw top histogram:
   for (i in 1:length(h.y1n)){
     if (h.y1n[i] < 1)
      polygon(c(rep(h.br[i], 2), rep(h.br[i+1], 2)),
        c(h.y1n[i], 1, 1, h.y1n[i]), col = col.hist)
   }

  # add counts to bins if required:
   if (count.hist == TRUE)
    for (i in 1 : length(h.x)){
     text(h.x[i], h.y1n[i], h.y1[i], cex = 1, pos = 1)
     text(h.x[i], h.y0n[i], h.y0[i], cex = 1, pos = 3)
    }

  # plot the axes of histograms:
   axis.hist <- function (h.y0, h.y1, scale.hist,
      las = las.h1){
    tope <- max(c(h.y0, h.y1))
    label.down <- c(0, (ceiling(tope/10))*5,
      (ceiling(tope/10))*10)
    label.up <- c((ceiling(tope/10))*10,
      (ceiling(tope/10))*5, 0)
    at.down <- label.down/(tope * scale.hist)
    at.up <- 1 - (label.up/(tope * scale.hist))
    at.hist <- c(at.down, at.up)
    label.hist <- c(label.down, label.up)
    axis(side = 4, at = at.hist, labels = label.hist,
      las = las)
    mtext("Frequency", side = 4, line = 2, cex = 1.5)
   }
   axis.hist(h.y0, h.y1, scale.hist)
   axis (side = 2, las = las.h1)
 }


# set the margins of plot area
old.mar <- par()$mar
par(mar = c(5.1,4.1,4.1,4.1))

# plot the combined graph
```

```
if (boxp == TRUE) logi.box(independ, depend)
if (boxp == FALSE) logi.scater(independ, depend)
if (type != "dit") logi.hist(independ, depend,...)
if (rug == TRUE) logi.rug (independ, depend)
logi.curve(independ, depend)
if (type == "dit") logi.dit(independ, depend)

# reset the margins to old margins
par(mar = old.mar)
}


# Example data, from library gravy of J. Oksanen

altitude <- c(930, 945, 955, 955, 960, 970, 990, 1000, 1000, 1005, 1010, 1010,
1015, 1015, 1020, 1020, 1020, 1030, 1030, 1030, 1030, 1030, 1035, 1045, 1050,
1050, 1050, 1060, 1065, 1065, 1065, 1070, 1070, 1075, 1080, 1080, 1080, 1085,
1090, 1090, 1090, 1090, 1095, 1100, 1100, 1100, 1100, 1100, 1110, 1110, 1110,
1110, 1120, 1120, 1120, 1120, 1120, 1120, 1120, 1125, 1130, 1130, 1130, 1130,
1130, 1130, 1135, 1135, 1140, 1140, 1140, 1140, 1140, 1140, 1140, 1140, 1150,
1150, 1160, 1160, 1160, 1160, 1165, 1170, 1170, 1170, 1175, 1180, 1180, 1180,
1180, 1180, 1185, 1190, 1190, 1190, 1195, 1200, 1200, 1205, 1210, 1210, 1215,
1215, 1215, 1220, 1220, 1220, 1220, 1220, 1220, 1225, 1230, 1230, 1235, 1240,
1240, 1250, 1250, 1250, 1250, 1250, 1250, 1255, 1255, 1255, 1255, 1260, 1260,
1260, 1265, 1265, 1270, 1270, 1270, 1270, 1275, 1275, 1275, 1275, 1275, 1275,
1280, 1285, 1285, 1290, 1290, 1290, 1300, 1300, 1300, 1310, 1310, 1310, 1330,
1350, 1355, 1360, 1365, 1365, 1365, 1365, 1370, 1370, 1370, 1370, 1380)


tree <- c(0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1,
1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1,
1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0)
```

Focus on Field Stations

University of Michigan Biological Station (UMBS)

Without the sign at the main entrance of the University of Michigan Biological Station UMBS), you might not suspect that this driveway leads to land that has been a research and teaching field station since 1909. And without a map, you might not have realized that during the last two miles of your drive you were already surrounded by the Station's property. The Biological Station manages 10,000 acres (4050 ha) bounded by undeveloped shoreline, including 9 km on Douglas Lake (15.2 km^2 area) and 2.5 km on Burt Lake (69.29 km^2) (Fig. 1).

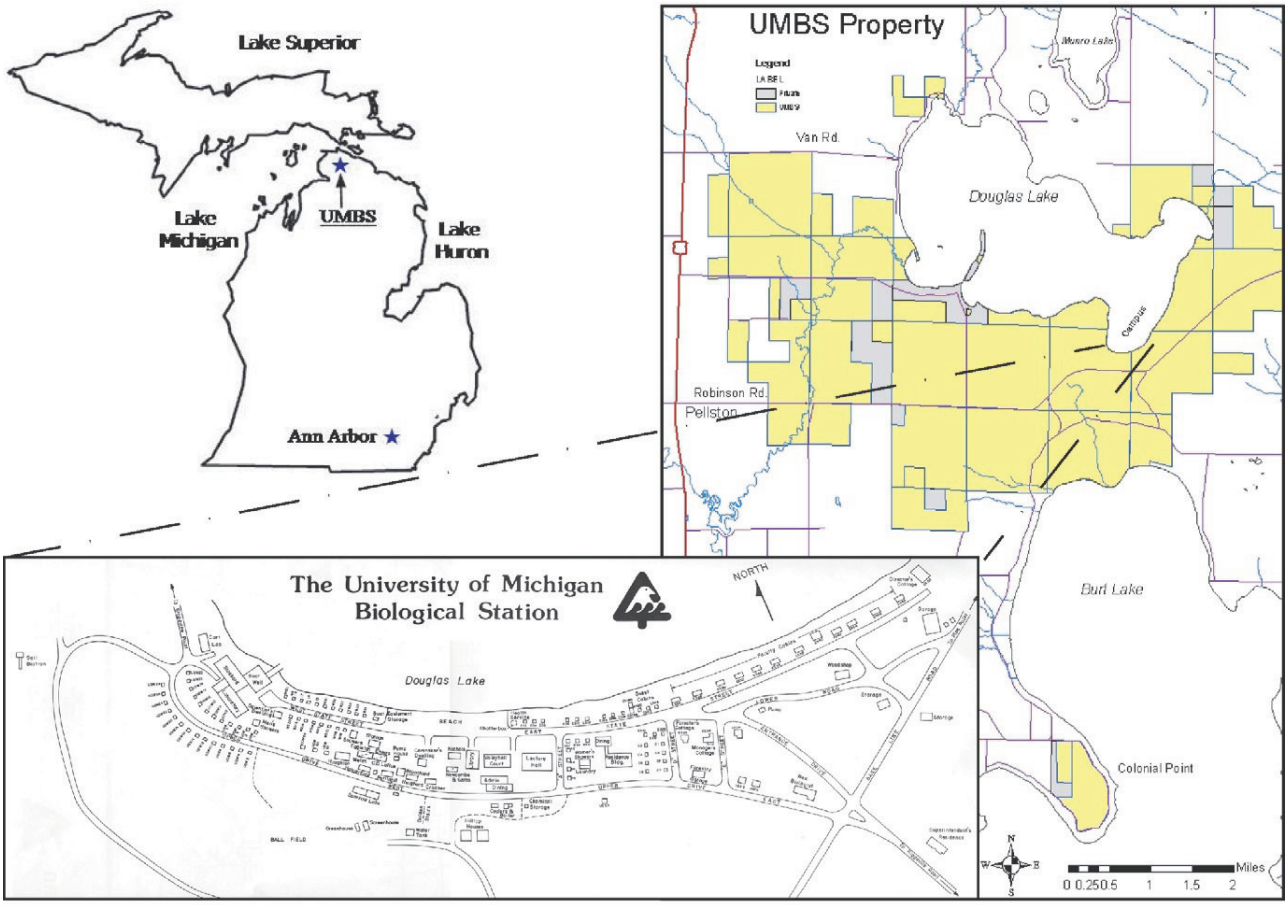


Fig. 1. Location of the University of Michigan Biological Station in northern Michigan. The principal land holdings (~10,000 acres [4050 ha]) of the UMBS are shown in yellow in the first inset. The campus (housing, laboratories, classrooms, laboratories, service buildings) is shown in the second inset. Sugar Island structures and land (~3,200 acres [~1300 ha]), about 60 miles [97 km] north) are not shown.

The holdings contain a rich diversity of natural habitats: extensive forests of pine, northern hardwoods, conifer swamps, and successional aspen stands, fields and meadows, pine plains, rivers, streams, and wetlands. Designated as a research and natural area available for use by students, faculty, and visiting researchers, public access is allowed, but off–road motorized vehicles are prohibited. Farther north, UMBS