11: Ordination

John H Maindonald

August 21, 2014

Ideas and issues illustrated by the graphs in this vignette

Ordination is a generic name for methods for providing a low-dimensional view of points in multi-dimensional space, such that "similar" objects are near each other and dissimilar objects are separated. The plot(s) from an ordination in 2 or 3 dimensions may provide useful visual clues on clusters in the data and on outliers. The methods described heere all use some form of multi-dimensional scaling (MDS)

1 Code for the Figures

```
fig9.1 <- function(){
  opar <- par(xpd=TRUE)
  if(!exists("aupoints")){
    cat("Trying to obtain audists from DAAG")
    if(!require(DAAG))return("'DAAG' must be installed")
    aupoints <- cmdscale(audists)
  }
    assign('aupoints', aupoints, pos=1)
    xlim <- range(aupoints[,1])
    xlim <- xlim + diff(xlim)*c(-0.05, 0.2)
    plot(aupoints, bty="n", xlab="", ylab="", xlim=xlim)
    labs <- rownames(aupoints)
    labpos <- rep(1, length(labs))
    labpos[labs%in%c("Cairns","Melbourne")] <- 3
    labpos[labs=="Canberra"] <- 4
    text(aupoints, labels=labs, pos=labpos, xpd=TRUE)
  par(opar)
}
```

```
fig9.2A <- function(){
    if(!require(DAAG))return("'DAAG' must be installed")
    if(!require(oz))return("Package 'oz' must be installed")
    if(!exists('aupoints'))aupoints <- cmdscale(audists)
    oz()
    points(aulatlong, col="red", pch=16, cex=1.5)
    latlong <- align2D(lat=aulatlong[,1], long=aulatlong[,2],
                       x1=aupoints[,1], x2 = aupoints[,2])
    x <- with(latlong, as.vector(rbind(lat, fitlat, rep(NA,10))))
    y <- with(latlong, as.vector(rbind(long, fitlong, rep(NA,10))))
    lines(x, y, col="gray40", lwd=3)
}
```

```
fig9.2B <- function(){
    if(!require(MASS))return("Package 'MASS' must be installed")
    if(!require(oz))return("Package 'oz' must be installed")
    aupoints.sam <- sammon(audists, trace=FALSE)
    oz()
    points(aulatlong, col="red", pch=16, cex=1.5)
    wt <- apply(as.matrix(audists), 1,function(x)sum(1/x[x>0]))
    latlong <- align2D(lat=aulatlong[,1], long=aulatlong[,2],
                       x1=aupoints.sam$points[,1],
                       x2 = aupoints.sam$points[,2], wts=wt)
    x <- with(latlong, as.vector(rbind(lat, fitlat, rep(NA,10))))
    y <- with(latlong, as.vector(rbind(long, fitlong, rep(NA,10))))
    lines(x, y, col="gray40", lwd=3)
}
```

```
fig9.2 <- function(){
  par(fig=c(0,1,0.5,1))
  fig9.2A()
  par(fig=c(0,1,0,0.5), new=TRUE)
  fig9.2B()
}
```

```
fig9.3A <- function(seed=47, xlab="Axis 1", ylab="Axis 2"){
    if(!require(DAAGbio))return("Package 'DAAGbio' must be installed")
    if(!require(ape))return("Package 'ape' must be installed")
    ## Calculate distances, using Kimura's K80 model
    primates.dist <- dist.dna(as.DNAbin(primateDNA), model="K80")
    primates.cmd <- cmdscale(primates.dist)
    eqscplot(primates.cmd, xlab=xlab, ylab=ylab, cex.lab=1.15)
    lefrt <- 2+2*(primates.cmd[,1] < mean(par()$usr[1:2]))
    text(primates.cmd[,1], primates.cmd[,2], row.names(primates.cmd),
         pos=lefrt)
}
```

```
fig9.3B <- function(seed=47, xlab="Axis 1", ylab="Axis 2"){
    if(!require(DAAGbio))return("Package 'DAAGbio' must be installed")
    if(!require(ape))return("Package 'ape' must be installed")
    if(!require(MASS))return("Package 'MASS' must be installed")
    primates.dist <- dist.dna(as.DNAbin(primateDNA), model="K80")
    primates.cmd <- cmdscale(primates.dist)
    primates.mds <- isoMDS(primates.dist, primates.cmd, k=2, trace=FALSE)
    eqscplot(primates.mds$points, xlab=xlab, ylab=ylab,
             cex.lab=1.15)
    lefrt <- 2+2*(primates.mds$points[,1] < mean(par()$usr[1:2]))
    text(primates.mds$points[,1], primates.mds$points[,2],
         row.names(primates.mds$points), pos=lefrt)
}
```

```
fig9.3 <- function(){
  opar <- par(fig=c(0,0.5,0,1), mar=c(3.1,3.1,1.6,0.1))
  fig9.3A()
  par(fig=c(0.5,1,0,1), new=TRUE)
  fig9.3B(ylab="")
  par(fig=c(0,1,0,1))
  par(opar)
}
```

```
fig9.4 <- function(){
    if(!require(DAAG))return("Package 'DAAG' must be installed")
    if(!require(MASS))return("Package 'MASS' must be installed")
    pacific.dist <- dist(x = as.matrix(rockArt[-c(47,54,60,63,92),
                         28:641]), method = "binary")
    sum(pacific.dist==1)/length(pacific.dist)
    ## Now check that in all columns at least one distance < 1
    symmat <- as.matrix(pacific.dist)
    checksum <- sum(apply(symmat, 2, function(x) sum(x<1)))
    checksum <- sum(apply(symmat, 2, function(x) sum(x<1)==0))
    print(c("No of cols where all distances are one"=checksum))
    pacific.cmd <- cmdscale(pacific.dist)
    pacific.mds <- isoMDS(pacific.dist, pacific.cmd, trace=FALSE)
    plot(pacific.mds$points)
}
```

2 Show the Figures

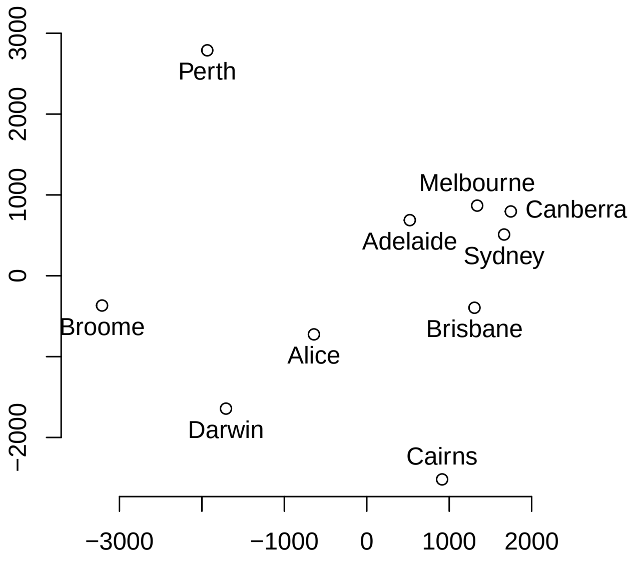
Unless doFigs is found in the workspace and is FALSE, then subject to checks that all necessary datasets and packages are available, the figures are now shown.

```
if(!exists("doFigs")) doFigs <- TRUE
```

```
pkgs <- c("DAAG","DAAGbio","MASS","oz","ape")
z <- sapply(pkgs, require, character.only=TRUE, warn.conflicts=FALSE)
if(any(!z)){
  notAvail <- paste(names(z)[!z], collapse=", ")
  print(paste("The following packages should be installed:", notAvail))
}
```

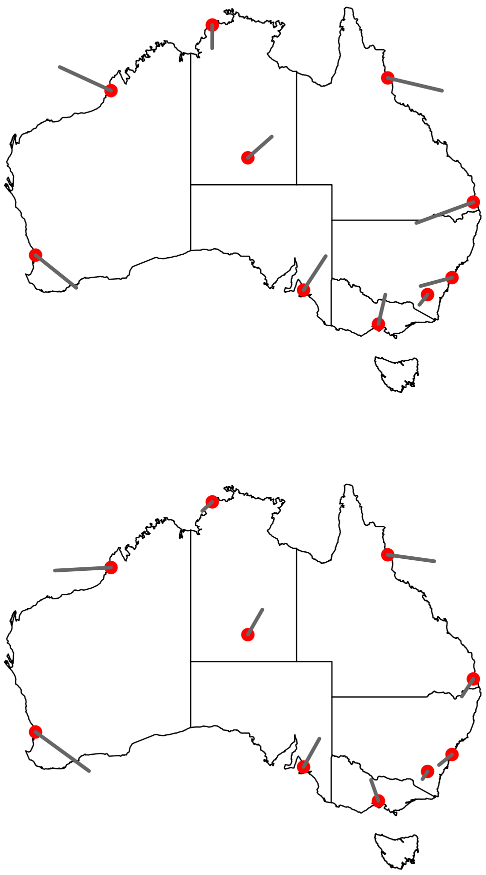
```
if(doFigs)fig9.1()

Trying to obtain audists from DAAG
```

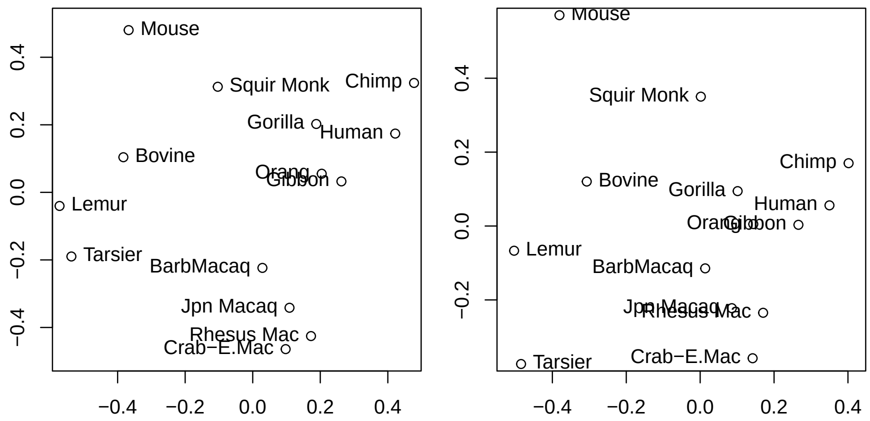


```
if(doFigs)if(!exists("aupoints"))
aupoints <- cmdscale(audists)
```

```
if(doFigs)fig9.2()
```

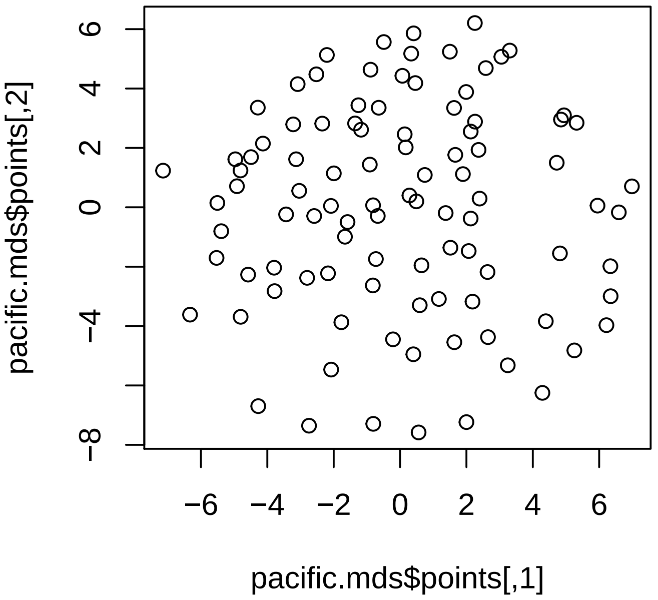


```
if(doFigs)fig9.3()
```



```
if(doFigs)fig9.4()

No of cols where all distances are one
                                     0
```


pacific.mds$points[,2]
pacific.mds$points[,1]