Alexey Shipunov

Marchine learning for zoologists and botanists

Kyoto University Seminar

2020

Contents

Seminar 1

1.1 Obtain tools

1. Use CRAN, download and run the latest installation package
2. Windows users: elect "SDI" mode; macOS users: try Terminal.app
3. Text editor: download and install Geany
4. "Visual Statistics. Use R!" book. Guides in Japanese on CRAN here: https://cran.r-project.org/doc/contrib/manuals-jp/

1.2 Load data

1. `q(save="no")`, quotes, brackets, spaces, letter case, argument `name=value`, why to answer "No"
2. `help(q)`, `example(boxplot)`, `help.search("histogram", lib.loc=.Library)`. To get help internally (*not* in browser), type: `options(help_type="text")`
3. Arrow Up and Tab keys
4. Working directory and `getwd()`
5. Load external data: locate, look, `read.table()` and `str()`
6. Data from spreadsheets: clipboard and tab-delimited files (macOS specifics), example is http://ashipunov.info/shipunov/school/biol_240/data/eq.xls

1.3 Process data

1. Dimension reduction via principal component analysis (PCA)
2. Plotting the two first principal components

1.4 Future

1. Date and time of the next seminar.
2. Comminucation: shipunov.alexey.7w@kyoto-u.ac.jp

Seminar 2

2.1 R scripting

1. First script: `hello.r`, how to save and run it with `source()`
2. Plots, save as PDF: open device, plot, close device
3. Second script: `iris.r` which plots PCA results

2.2 Projection, deconvolution and intermediate methods of dimension reduction

1. PCA restrictions: it is based on projection, and uses measurement data with linear assumptions.
2. Deconvolutions: isomap (`Rdimtools::do.isomap()`)
3. Intermediate methods: UMAP (`uwot::umap()`)

2.3 Homework

1. Debugging: how to make the `scriptb.r` (http://ashipunov.info/shipunov/school/biol_240/scriptb.r) work

2.4 Future

Distances, MDS, clustering, LDA, MANOVA, recursive partitioning, bagging and boosting, rules methods, SVM, neural networks, semi-supervised methods. Geometric morphometry in R.

Seminar 3

3.1 R skills

1. Run scripts which are available online.

```
source("http://ashipunov.info/shipunov/school/biol_240/ncov.r", echo=TRUE)
plot(log(confirmed) ~ dt, data=cc, type="b", xlab="Date",
 ylab="Confirmed cases (log)")
```

2. Homework: how to make the `scriptb.r` (http://ashipunov.info/shipunov/school/biol_240/scriptb.r) work?

```
## will produce error:
## source("http://ashipunov.info/shipunov/school/biol_240/scriptb.r", echo=TRUE
## t1 <- read.table("trees_m.txt") # this the place of error
## file.show("trees_m.txt") # look on data
## str(t1) # look on result (1)
## head(t1) # look on result (2)
## look on script:
## url.show("http://ashipunov.info/shipunov/school/biol_240/scriptb.r")
## this is how to do it properly:
trees.m <- data.frame(trees[1]*2.54, trees[2]*30.48, trees[3]*(30.38^3))
write.table(trees.m, file="trees_m.txt", row.names=FALSE, quote=FALSE)
t2 <- read.table("trees_m.txt", h=TRUE)
t2.scaled <- scale(t2)
boxplot(t2.scaled)
## do not forget to remove the file (but be careful with this command!)
## unlink("trees_m.txt")
```

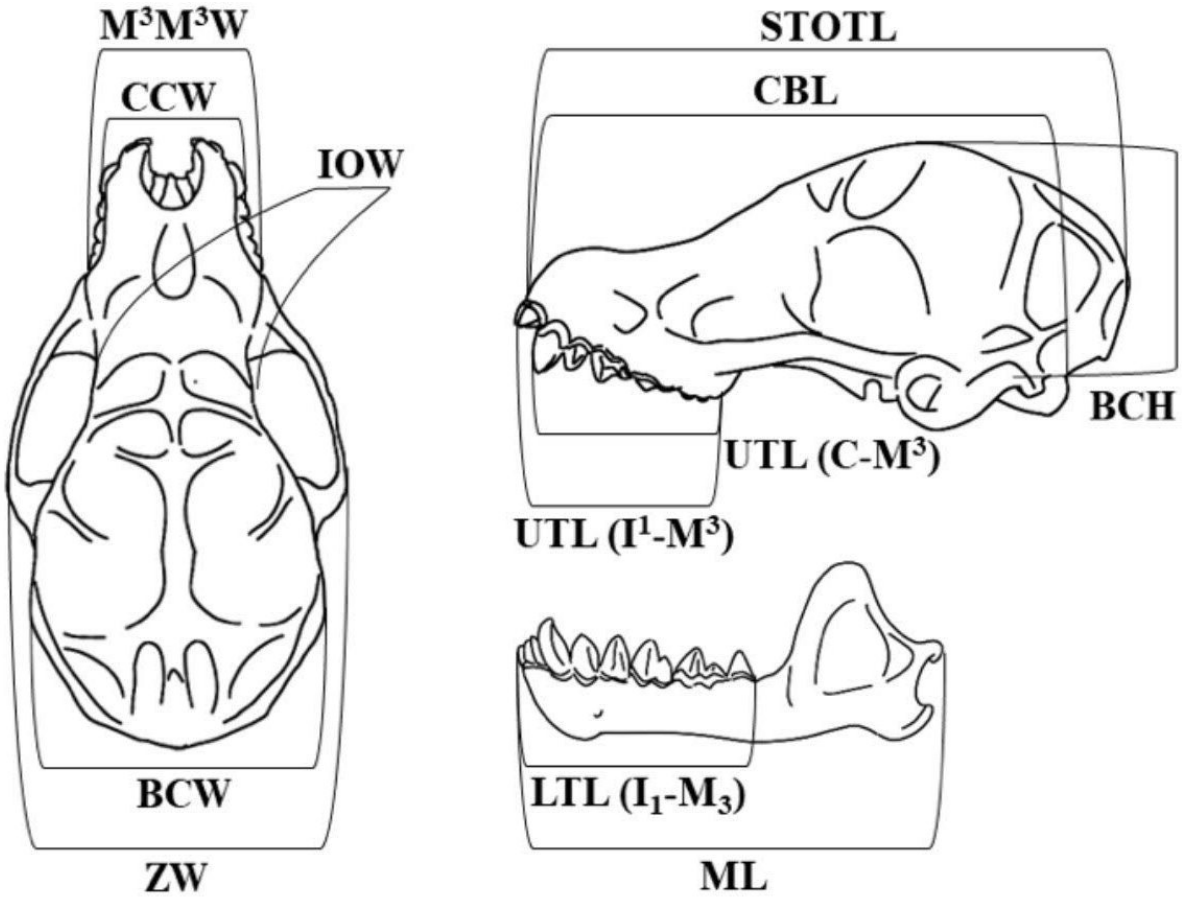
3. More about PCA: biplot, convex hulls and overlap

```
iris.p <- prcomp(iris[, -5], scale=TRUE)
iris.p # loadings (importances of variables)
summary(iris.p) # total variance explained (importances of components)
biplot(iris.p, xpd=TRUE) # shows both original and PCA variables
iris[16, ] # row number 16

library(shipunov) # install also 'PBSmapping' package
plot(iris.p$x, col=iris$Species)
iris.h <- Hulls(iris.p$x[, 1:2], groups=iris$Species)
```

```
summary(Overlap(iris.h))
iris.h <- Hulls(iris.p$x[, 1:2], groups=iris$Species)
summary(Overlap(iris.h))
```

4. How to use *Murina hilgendorfi* data.



Craniodental and mandibular measurements by caliper (linear measurement)

```
## 'sample_murina_hilgendorfi.xls' converted into tab-delimited text file
mh <- read.table(
 "http://ashipunov.info/shipunov/school/biol_240/sample_murina_hilgendorfi.txt",
 h=TRUE, sep="\t")
str(mh)
Str(mh) # useful command: shows number of variable and presence of missing data
sapply(mh[, -(1:4)], Normality) # some variab;es are not normal
mh.cor <- cor(mh[, -(1:4)], method="spearman") # correlation matrix, non-parametric
Pleiad(mh.cor, corr=TRUE, breaks=3) # correlogramm

mh.p <- prcomp(mh[, -(1:4)], scale=TRUE)
palette(rainbow(nlevels(mh$locality)))
plot(mh.p$x, col=mh$locality, pch=as.character(mh$locality))
mh.ph <- Hulls(mh.p$x, groups=mh$locality)
mh.ov <- Overlap(mh.ph)
summary(mh.ov) # calculates overlap of each with all others
palette("default")
Biarrows(mh.p$x, scale(mh[, -(1:4)]), ar.col=1, tx.col=1) # biplot-like

library(Rdimtools)
```

```
mh.i <- do.isomap(scale(mh[, -(1:4)]), type=c("enn", 6))
palette(rainbow(nlevels(mh$locality)))
plot(mh.i$Y, col=mh$locality, pch=as.character(mh$locality))
mh.ih <- Hulls(mh.i$Y, groups=mh$locality)
palette("default")
Biarrows(mh.i$Y, scale(mh[, -(1:4)]), ar.col=1, tx.col=1)
summary(Overlap(mh.ih))

library(uwot)
mh.u <- umap(scale(mh[, -(1:4)]))
oldpal <- palette(rainbow(nlevels(mh$locality)))
plot(mh.u, col=mh$locality, pch=as.character(mh$locality))
mh.uh <- Hulls(mh.u, groups=mh$locality)
palette(oldpal)
Biarrows(mh.u, scale(mh[, -(1:4)]), ar.col=1, tx.col=1)
summary(Overlap(mh.uh))
```

3.2 Layout Learning

1. Distances: Euclidean, Gower, Jaccard

```
iris.d <- dist(iris[, -5], method="euclidean")
library(shipunov)
iris.g <- Gower.dist(iris) # note that I did not remove 5th column
bin <- +t(moldino > 0) # make binary (occurrence) dataset
library(vegan)
bin.j <- vegdist(bin, dist="jaccard")
```

2. Multidimensional scaling (MDS = PCoA), biplot, surrogate variance and loadings

```
iris.c1 <- cmdscale(iris.d)
plot(iris.c1, col=iris$Species)
Hulls(iris.c1, groups=iris$Species)
MDSv(iris.c1) # importances of dimensions (surrogate explained variance)
Biarrows(iris.c1, scale(iris[, -5])) # biplot-like but for MDS

iris.c2 <- cmdscale(iris.g)
plot(iris.c2, col=iris$Species)
Hulls(iris.c2, groups=iris$Species) # much better -- because we told species

library(MASS)
iris.mds <- isoMDS(dist(iris[, -5]) + 1e-9) # non-metric MDS
cor(iris[, 1:4], iris.mds$points) # variable importances ("loadings")
library(shipunov)
(vv <- MDSv(iris.mds$points)) # dimension importance ("explained variance")
xxlab <- paste0("Dim 1 (", round(vv[1], 2), "%)")
yylab <- paste0("Dim 1 (", round(vv[2], 2), "%)")
abb <- abbreviate(iris$Species, 1, method="both.sides")
plot(iris.mds$points, pch=abb, xlab=xxlab, ylab=yylab)
Biarrows(iris.mds$points, iris[, -5]) # biplot for MDS
iris.mds.h <- Hulls(iris.mds$points, groups=iris[, 5], usecolors=rep(1, 3), lty=2)
```

```
summary(Overlap(iris.mds.h))

library(vegan)
mh.d <- dist(mh[, -(1:4)])
# # run several times, not always converged:
mh.mds <- metaMDS(mh.d, distance="euclidean")
palette(rainbow(nlevels(mh$locality)))
plot(mh.mds$points, col=mh$locality, pch=as.character(mh$locality))
mh.h2 <- Hulls(mh.mds$points, groups=mh$locality)
summary(Overlap(mh.h2))
palette("default")
Biarrows(mh.mds$points, mh[, -c(1:4)], ar.col=1, tx.col=1, lty=2)
```

3. Inferential PCA and MDS

```
library(vegan)
anosim(iris.c1, iris$Species, dist="euclidean")
anosim(iris.c2, iris$Species, dist="euclidean")
anosim(iris.p$x[, 1:2], iris$Species, dist="euclidean")
```

4. Hierarchical clustering and linkage; bootstrap

```
iris.10 <- iris[c(rep(0, 9), 1) > 0, ] # select every 10th row
iris.10d <- dist(iris.10[, -5])
iris.10dh <- hclust(iris.10d, method="ward.D")
plot(iris.10dh, labels=iris.10$Species)

library(shipunov)
iris.10b <- Bclust(iris.10[, -5])
plot(iris.10b$hclust, labels=iris.10$Species)
Bclabels(iris.10b$hclust, iris.10b$values, col="red", pos=3, offset=0.1)

## clustering overlaps
mh.ov # all overlaps
mh.ov[is.na(mh.ov)] <- 0
mh.od <- as.dist(1-mh.ov) # use overlap as distance
plot(hclust(mh.od)) # which groups are closer by overlap
```

5. How to "reverse" trees (and make super-trees)

```
library(shipunov)
iris.h1 <- hclust(dist(iris[, -5]), method="ward.D")
iris.h2 <- hclust(dist(iris[, -5]), method="single")
iris.b1 <- MRH(iris.h1) # raw data from trees
iris.b2 <- MRH(iris.h2)
iris.12 <- cbind(iris.b1, iris.b2) # merge trees data
plot(cmdscale(dist(iris.12)), col=iris$Species) # result
```

6. Partitioning with desired number of clusters; fuzzy methods

```
iris.k <- kmeans(iris[, -5], centers=3)
plot(iris.p$x, col=iris$Species, pch=iris.k$cluster)
library(shipunov)
Misclass(iris$Species, iris.k$cluster, best=TRUE)

mh.k <- kmeans(mh[, -(1:4)], centers=5)
palette(rainbow(nlevels(mh$locality)))
plot(mh.p$x, col=mh$locality, pch=mh.k$cluster)
palette("default")
## Misclass(mh$locality, mh.k$cluster, best=TRUE) # does not work with 9 classes...
new <- as.character(mh$locality) # colnvert locality to character
new2 <- ifelse(!new %in% c("W", "I", "L", "U"), "C", new) # 5 classes only
Misclass(new2, mh.k$cluster, best=TRUE) # insteresting!

library(cluster)
iris.f <- fanny(iris[, -5], k=3)
iris.fzz <- apply(iris.f$membership, 1, var) # fuzziness
tt <- quantile(iris.fzz, 0.25) # threshold
iris.p <- prcomp(iris[, -5], scale=TRUE) # we need it for plotting
plot(iris.p$x, col=iris[, 5], pch=ifelse(iris.fzz < tt, 1, 19))
```

7. *Ad hoc* partitioning: mean-shift and others

```
library(meanShiftR)
iris.m <- meanShift(as.matrix(iris[, -5]))
plot(iris.p$x, col=iris$Species, pch=iris.m$assignment)
library(shipunov)
Misclass(iris.m$assignment, iris$Species, best=TRUE)

mh.m <- meanShift(as.matrix(mh[, -(1:4)]))
palette(rainbow(nlevels(mh$locality)))
plot(mh.p$x, col=mh$locality, pch=mh.m$assignment) # not useful...
palette("default")
```

3.3 Future

Machine learning in the strict sense: supervised methods. LDA, MANOVA (including non-parametric), recursive partitioning, bagging and boosting, rules methods, k-NN, SVM, neural networks, semi-supervised methods. Geometric morphometry in R.

Seminar 4

4.1 R skills

```
## I used feedback from the previous seminar to improve Misclass()
## update.packages("shipunov")
```

4.2 Supervised methods

```
## first, we define training and (fake) testing groups:
iris.train <- iris[seq(1, nrow(iris), 5), ]
iris.unknown <- iris[-seq(1, nrow(iris), 5), ]
```

1. Linear Discriminate Analysis (LDA)

   ```
   library(MASS)
   iris.lda <- lda(Species ~ ., data=iris.train)
   iris.ldap <- predict(iris.lda, iris.unknown[, -5])
   library(shipunov)
   Misclass(iris.unknown$Species, iris.ldap$class)
   ```

2. Multivariate ANOVA

   ```
   summary(manova(as.matrix(iris[, -5]) ~ iris$Species), test="Wilks")
   library(vegan)
   adonis(as.matrix(iris[, 1:4]) ~ iris$Species)
   ```

3. Recursive partitioning

   ```
   library(rpart)
   iris.rpart <- rpart(Species ~ ., data=iris.train)
   iris.rpartp <- predict(iris.rpart, iris.unknown[, -5], type="class")
   Misclass(iris.unknown$Species, iris.rpartp)
   plot(iris.rpart); text(iris.rpart, xpd=TRUE)
   ```

4. Bagging: Random Forest

   ```
   library(ranger)
   iris.rg <- ranger(Species ~ ., data=iris.train, importance="impurity")
   iris.rgp <- predict(iris.rg, iris.unknown[, -5])
   library(shipunov)
   Misclass(iris.unknown$Species, iris.rgp$predictions)
   iris.rg$variable.importance
   ```

5. Boosting: extreme gradient boosting

```
library(gbm)
iris.gbm <- gbm(Species ~ ., data=rbind(iris.train, iris.train)) # make bigger
iris.gbmp1 <- predict(iris.gbm, iris.unknown[, -5], n.trees=iris.gbm$n.trees)
iris.gbmp2 <- apply(iris.gbmp1, 1,
 function(.x) colnames(iris.gbmp1)[which.max(.x)]) # membership trick
library(shipunov)
Misclass(iris.unknown$Species, iris.gbmp2)
```

6. Naïve Bayes and how to plot the learning

```
library(e1071)
iris.nb <- naiveBayes(Species ~ ., data=iris.train)
iris.nbp <- predict(iris.nb, iris.unknown[, -5])
library(shipunov)
Misclass(iris.unknown$Species, iris.nbp)

iris.p <- prcomp(iris[, -5], scale=TRUE)$x[, 1:2]
sel <- 1:nrow(iris) %in% seq(1, nrow(iris), 5)
plot(iris.p, col=iris$Species, pch=ifelse(sel, 19, 1), main="naiveBayes")
iris.nb2 <- naiveBayes(Species ~ ., data=cbind(iris[5], iris.p)[sel, ])
Gradd(iris.nb2, iris.p[sel, ], what="lines")
```

7. Nearest neghbor methods

```
library(class)
iris.knn.pred <- knn(train=iris.train[, -5],
 test=iris.unknown[, -5], cl=iris.train[, 5], k=5)
library(shipunov)
Misclass(iris.knn.pred, iris.unknown[, 5])

iris.bootknn <- BootKNN(iris[, -5], iris[, 5])
st <- apply(iris.bootknn, 1, function(.x) var(as.numeric(as.factor(.x))))
plot(prcomp(iris[, -5])$x, col=iris$Species, pch=ifelse(st == 0, 19, 1))

cl1 <- iris$Species
sam <- c(rep(0, 4), 1) > 0
cl1[!sam] <- NA

for (d in (5:14)/100) {
iris.pred <- DNN(dst=dist(iris[, -5]), cl=cl1, d=d)
cl1[is.na(cl1)] <- iris.pred
}
table(cl1, useNA="ifany")
Misclass(iris$Species, cl1)

cl2 <- iris$Species
iris.d <- Gower.dist(iris[, -5])
iris.proximity <- t(DNN(dst=iris.d, cl=cl2, k=5, details=TRUE, self=TRUE))/5
head(iris.proximity)
```

8. Support Vector Machines

```
library(e1071)
iris.svm <- svm(Species ~ ., data=iris.train)
iris.svmp <- predict(iris.svm, iris.unknown[, -5])
library(shipunov)
Misclass(iris.unknown$Species, iris.svmp)
```

9. Neural Networks

```
library(nnet)
iris.nn <- nnet(Species ~ . , data=iris.train, size=4)
iris.nnp <- predict(iris.nn, iris.unknown[, -5], type="class")
library(shipunov)
Misclass(iris.unknown$Species, iris.nnp)
library(NeuralNetTools)
oldpar <- par(mar=c(0, 2, 0, 1), xpd=TRUE)
plotnet(iris.nn)
par(oldpar)
```

4.3 Mixed methods

Cluster sharpening

```
library(kpeaks)
nn <- max(findk(iris[, -5])$pcounts)
library(ksharp)
iris.km <- kmeans(iris[, -5], centers=nn)
Misclass(iris$Species, iris.km$cluster, best=TRUE)
names(iris.km$cluster) <- row.names(iris)
iris.ksharp <- ksharp(iris.km, data=iris[, -5])
library(shipunov)
Misclass(iris$Species, iris.ksharp$cluster, ignore=0, best=TRUE)

iris.kx <- list(cluster=as.numeric(iris$Species))
names(iris.kx$cluster) <- row.names(iris)
iris.ksharpx <- ksharp(iris.kx, data=iris[, -5])
plot(prcomp(iris[, -5])$x, col=iris$Species,
 pch=ifelse(iris.ksharpx$cluster == 0, 1, 19))
```

4.4 Geometric morphometrics

Package `geomorph` example

```
library(geomorph)
## # download: http://ashipunov.me/shipunov/school/biol_240/061.jpg
## # download: http://ashipunov.me/shipunov/school/biol_240/046.jpg
## # download: http://ashipunov.me/shipunov/school/biol_240/650.jpg
## # check that all these files present in working directory
## digitize2d(filelist=c("061.jpg", "406.jpg", "650.jpg"),
```

```
## nlandmarks=4, tpsfile="046.tps")
## readland.tps("046.tps")

am <- read.table("http://ashipunov.me/data/bigaln2.txt", as.is=TRUE, head=TRUE)
ag <- readland.tps("http://ashipunov.me/data/bigaln.tps", specID="imageID")
ag.gpa <- gpagen(ag) # Generalized Procrustes Analysis and projection
ag.pca <- gm.prcomp(ag.gpa$coords)$x # PCA in tangent space
veins <- as.numeric(cut(am$N.ZHILOK, 2))
plot(ag.pca, col=veins, main="Shape vs. number of lateral veins")

v1 <- mshape(ag.gpa$coords[, , veins == 1])
v2 <- mshape(ag.gpa$coords[, , veins == 2])
all <- mshape(ag.gpa$coords)
ag.links <- matrix(c(1, rep(c(2:7, 12:8), each=2), 1), ncol=2, byrow=TRUE)
ag.links # matrix of links
old.par <- par(mfrow=c(1, 2), mar=rep(0, 4)) # side by side
plotRefToTarget(v1, all, links=ag.links)
plotRefToTarget(v2, all, links=ag.links)
par(old.par)
```