Schools detection using the `R` package `acoustic` to control Echoview via COM

Lisa-Marie Harrison and Martin Cox

January 13, 2015

This vignette provides an example of preparing and importing multiple transects of acoustic data into Echoview using the `R` package `acoustic`. It is necessary to run this file before running the Schools Detection or Biomass Estimation examples. It is assumed that the user is already familiar with Echoview.

To run this example, you will need the following files:

1. **Echosounder data:** Simrad EK60 RAW data collected during the KAOS voyage.

2. **An Echoview template file:** An Echoview (.EV) file containing the virtual variables used during data processing and must be copied to the Echoview templates folder.

3. **Echoview calibration file:** The Echoview .ECS file containing the EK60 calibration parameters.

4. **Start and end file regions:** Start and end times of each transect stored as an Echoview region file.

All these files are available from ... and should be downloaded to the data directory specified in the `dd` object. If you want to reproduce this vignette, you should assign the location of the KAOS example data to dd.

First, set the working directory:

```
> dd='C:/Users/Lisa/Desktop/KAOS'
```

1 Loading RAW data into Echoview

In this section we load the necessary packages into the `R` workspace, open a COM connection between `R` and Echoview, then populate the Echoview template file with RAW Simrad EK60 data files.

```
> library(acoustic)
```

The EV file `KAOStemplate.EV` **must be copied into the c:/Program Files (x86)/Myriax/Echoview/Echoview6/Templates/directory.** Open a COM connection between `R` and `Echoview`:

```
> EVAppObj <- COMCreate('EchoviewCom.EvApplication')
```

Now we get a vector of the RAW data files to add to the Echoview template:

```
> pathAndFn=list.files(paste(dd,'raw',sep='/'),full.names=TRUE)
```

We remove any '.evi' files from the `pathAndFn`:

```
> eviLoc=grep('.evi',pathAndFn)
> if(length(eviLoc)>0) pathAndFn=pathAndFn[-eviLoc]
```

Important: Any filenames that are passed via COM must contain the full directory path.

2 Populate an Echoview template

Now that we have a vector of RAW data file locations, we can populate the Echoview template file specified in `EVCreateNew(templateFn)`.

```
> EVFile <- EVCreateNew(EVAppObj=EVAppObj,
+    templateFn=paste(dd, "/KAOStemplate.EV", sep = ""),
+    EVFileName=paste(dd,'kaosAll.ev',sep='/'),
+    filesetName="038-120-200",
+    dataFiles=pathAndFn)$EVFile
```

We can also populate a template in a loop, so if there are multiple transects, each containing a large amount of RAW data, we can create an EV file for each transect. To help illustrate this, there is a .csv file, `vignette_file_list_subset.csv`, included in the example data that specifies which transect the RAW data files are assigned to. First of all, we load this file into the `R` workspace:

```
> tF <- read.csv(paste(dd,'vignette_file_list_subset.csv',sep='/'))
> head(tF)
```

```
  transectNumber                        filename
1              1 L0055-D20030115-T171028-EK60.raw
2              1 L0055-D20030115-T182914-EK60.raw
3              2 L0056-D20030118-T000714-EK60.raw
4              3 L0056-D20030118-T131501-EK60.raw
5              3 L0056-D20030118-T143357-EK60.raw
6              3 L0056-D20030118-T155252-EK60.raw
```

Next, we append the raw data directory path to the raw data filenames:

```
> tF$filename <- paste(dd,'raw',tF$filename,sep='/')
```

Now we will loop over the transect file and create one EV file per transect. A calibration file is added to each .EV file. Regions definitions files for off transect times are also imported, which will remove off effort time from the analysis.

```
> uniqueTransect=unique(tF$transectNumber)
> for(i in 1:length(uniqueTransect)) {
+
+   EVFile <- EVCreateNew(EVAppObj=EVAppObj,
+   templateFn=paste(dd, "KAOStemplate.EV", sep = "/"),
+   EVFileName=paste(dd,'/kaos-transect-',i,'.ev',sep=''),
+   filesetName="038-120-200",
+   dataFiles=as.character(tF$filename[tF$transectNumber==uniqueTransect[i]]),
+   CloseOnSave = FALSE)$EVFile
+
+     #add a calibration file
+   EVAddCalibrationFile(EVFile = EVFile, filesetName = "038-120-200", calibrationFile = pas
+
+     #get a list of the regions definitions files to import and import individually
+   off_transect_files <- list.files(paste(dd, "/off transect regions", sep = ''),
+                                 full.names = T)
+   for (j in 1:length(off_transect_files)) {
+     EVImportRegionDef(EVFile, off_transect_files[j], paste("region_", j, sep = ""))
+   }
+
+   #add an EV line object and rename to 'seabed line'
+   evLine <- EVImportLine(EVFile, pathAndFn = 'C:/Users/Lisa/Desktop/KAOS/lineKAOS.evl')$li
+   EVRenameLine(EVFile = EVFile, evLine = evLine, newName = "seabed line")
+
+   #save the open .EV file
+   EVSaveFile(EVFile = EVFile)
+
+   #close the current transect
+   EVCloseFile(EVFile = EVFile)
+
+   }
>
```

This method allows subsets of a full survey to be processed separately which is useful for large data sets. Rather than running transects separately as shown above, the same method could also be used to run each survey day separately. The Biomass Estimation and Schools Detection vignettes follow on from this point and require this example to have been run first.