An introduction to the wraspp package

Lasse Bombien   Raphael Winkelmann

July 22, 2013

This document is meant to be an introduction to the wrassp package. wrassp is a wrapper to Michel Scheffer's libassp (Advanced Speech Signal Processor). The libassp library aims at providing functionality for handling speech signal files in most common audio formats and for performing analyses common in phonetic science/speech science. This includes the calculation of formants, fundamental frequency, root mean square, auto correlation, a variety of spectral analyses, zero crossing rate, filtering etc. This wrapper library for R exposes a large subset of the signal processing functions to R in a (hopefully) user friendly manner.

Let's get started by locating some example material distributed with the package.

```r
## load the package
library(wrassp)

## some audio files come along with the package
wavpath <- system.file("extdata", package = "wrassp")
wavfiles <- list.files(wavpath, pattern = glob2rx("*.wav"), full.names = TRUE)

## now we have some files to play with
```

One of the aims of wrassp is to provide mechanisms to handle speech related files such as sound files and parametric data files. wrassp therefore comes with a class called AsspDataObj which does just that.

```r
## load an audio file, e.g. the first one in the list above
au <- read.AsspDataObj(wavfiles[1])
class(au)

## [1] "AsspDataObj"

print(au)

## Assp Data Object of file /home/lasse/R/i686-pc-linux-gnu-library/3.0/wrassp/extdata/lbo001.wav.
## Format: WAVE (binary)
## 19983 records at 16000 Hz
## Duration: 1.248938 ms
## Number of tracks: 1
##   audio (1 fields)
```

au is an object of the named class and, using print, we can get some information on the object, such as it's sampling rate, its duration and what kind of data are stored in what form. Since the file we loaded is audio only, the object contains exactly one track. And since it's a mono file, the track only has one field. We will later encounter different types of data with more than one track and more fields.

There are more ways to extract attributes from the object, such as duration, sampling rate and the number of records:

```
## duration
dur.AsspDataObj(au)

## [1] 1.249

## sampling rate
rate.AsspDataObj(au)

## [1] 16000

## number of records/samples
numRecs.AsspDataObj(au)

## [1] 19983
```

An important property of AsspDataObj is of course that it contains data tracks, or at least one data track. As mentioned above, the currently loaded object contains a single mono audio track. Accessing the data is easy: AsspDataObj stores data in simple matrices, one matrix for each track. Broadly speaking, AsspDataObj is nothing but a list of at least one matrices. All of them have the same number of rows (number of records) but each can have a different number of columns (number of fields). Each track has a name and we can access the track using that name.

```
tracks.AsspDataObj(au)

## [1] "audio"

## or
names(au)

## [1] "audio"

head(au$audio)

##      [,1]
## [1,]    5
## [2,]   -2
## [3,]   17
## [4,]   -5
## [5,]   -5
## [6,]   -2
```

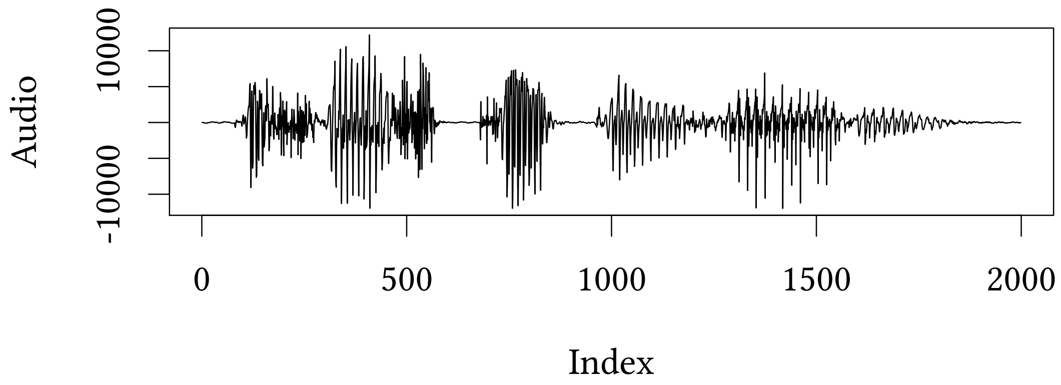

Figure 1: A simple way of plotting data

```r
## and we can of course also plot this way (only every tenth element)
plot(au$audio[c(TRUE, rep(FALSE, 9))], ty = "l", ylab = "Audio")
```