Extraction of acoustic information using Praat scripting

Dr. Christian DiCanio

University at Buffalo, Department of Linguistics

http://www.acsu.buffalo.edu/~cdicanio/

What do you know already? A lot!

1. How to use Praat via its GUI and create textgrids.
2. Variable types: numeric vs. string$
3. Loop types – for loops; if/else loops
4. Defining paths/directories
5. Querying information about imported files
6. Examining output from scripts
7. Cleaning up the object window

...And many other things, thanks to Thea and Chris!

Where are we going today?

1. We’re going to incorporate some of what we’ve learned to examine acoustic data from sound files.
2. We’ll read in sound files, extract acoustic information, and then print this to an output text file.
   a) Extract duration from intervals in a sound file (static acoustic measure)
   b) Extract formant values from intervals in a sound file (dynamic acoustic measure)

I: The dialog box

- It is useful to keep track of variables related to the files you are importing into Praat with a dialog box.

- In Praat, we use *form* and *endfor* for this. String variables are preceded by *sentence* and numeric variables by *positive*.

```
form Extract duration from labelled points
   sentence Name_of_directory:
/Teaching/AP_Lab/Praat_workshop_Thea_Chris_Me/te
sting/
   sentence Name_of_object: R1_0001
   positive Labelled_tier_number 2
   sentence Label a
   sentence Results Duration_data
endform
```

Variables must begin in uppercase here, but must always appear in lowercase in your script. The colon is ignored.

- The script *Get_duration_beginners_script.praat* contains the dialog box from the previous slide.

- Each time you run the script, you can set different values for the variables of interest or for the output file where our acoustic data will be stored.

II. Writing to text files

- Text files (.txt) are an extremely useful format for saving the output from running scripts. If they are tab-separated or comma-separated, they're directly readable by spreadsheet software, R, or other programs.

- There are two types of commands in Praat to work with external text files – commands for creating the labels in the text file and commands for appending actual values from the output of the script.

- The command *fileappend* creates a text file with the name we specified in the dialog box. We set 4 columns and separate each with a tab command – ‘tab$’

```
fileappend
'name_of_directory$''results$'.txt
Filename'tab$'Label'tab$'Duration'tab$'Int
ensity'newline$'
```

- But separately, inside a *for*-loop where we’ll get values, we need to write actual values to the columns of the text file.

```
fileappend
'name_of_directory$''results$'.txt
'name_of_object$''tab$''label$''tab$''dur'
'tab$''int''newline$'
```

All variables MUST be surrounded by single apostrophes now.

III. Variables vs. selected object window items

- We have to distinguish between variables in the script, which are either numeric values or strings and items in the object window which we have given variable names.
- The first needs no quotes, but the second requires it, but only when you select it.

```
Read from file...
'name_of_directory$''name_of_object$'.TextGrid
textGridID = selected("TextGrid")
num_labels = Get number of intervals...
labelled_tier_number
```

textgridID is the name of a file, so we refer to it in the script later as ‘textgridID’

num_labels is a variable, so we do not need quotes around it.

- In practice this looks like the following:

```
select ‘soundobject’
variablename = some command goes here for examining the acoustics of the sound

-or-

select ‘soundobject’
plus ‘textgridobject’
variablename = some command goes here for extracting parts of a sound with labels in the textgrid
```

IV. Extracting acoustic data

There are a couple ways to extract acoustic data from the speech signal in Praat.

1. Extract a small chunk of speech to the objects window, then iterate across the chunks (our duration script here).
2. Instead of chopping up the sound file, you can query the time domain using the textgrid boundaries and then extract values from portions of each, entire sound file.

- The first approach works fine if you want static acoustic values (duration, mean intensity, etc). You’re essentially just querying each chunk’s acoustics.

- It will *NOT* work well for examining values that vary over time. Why? Praat’s algorithms need to start tracking acoustic values before the domain of interest (LPC, autocorrelation, RMS).

- You need to get “a running start” to accurately estimate formant values, f0, or intensity as they vary in time. This favors the second approach since we are able to set larger windows than the domain we extract values from. (More on this later.)

Exercise 1: Examining duration

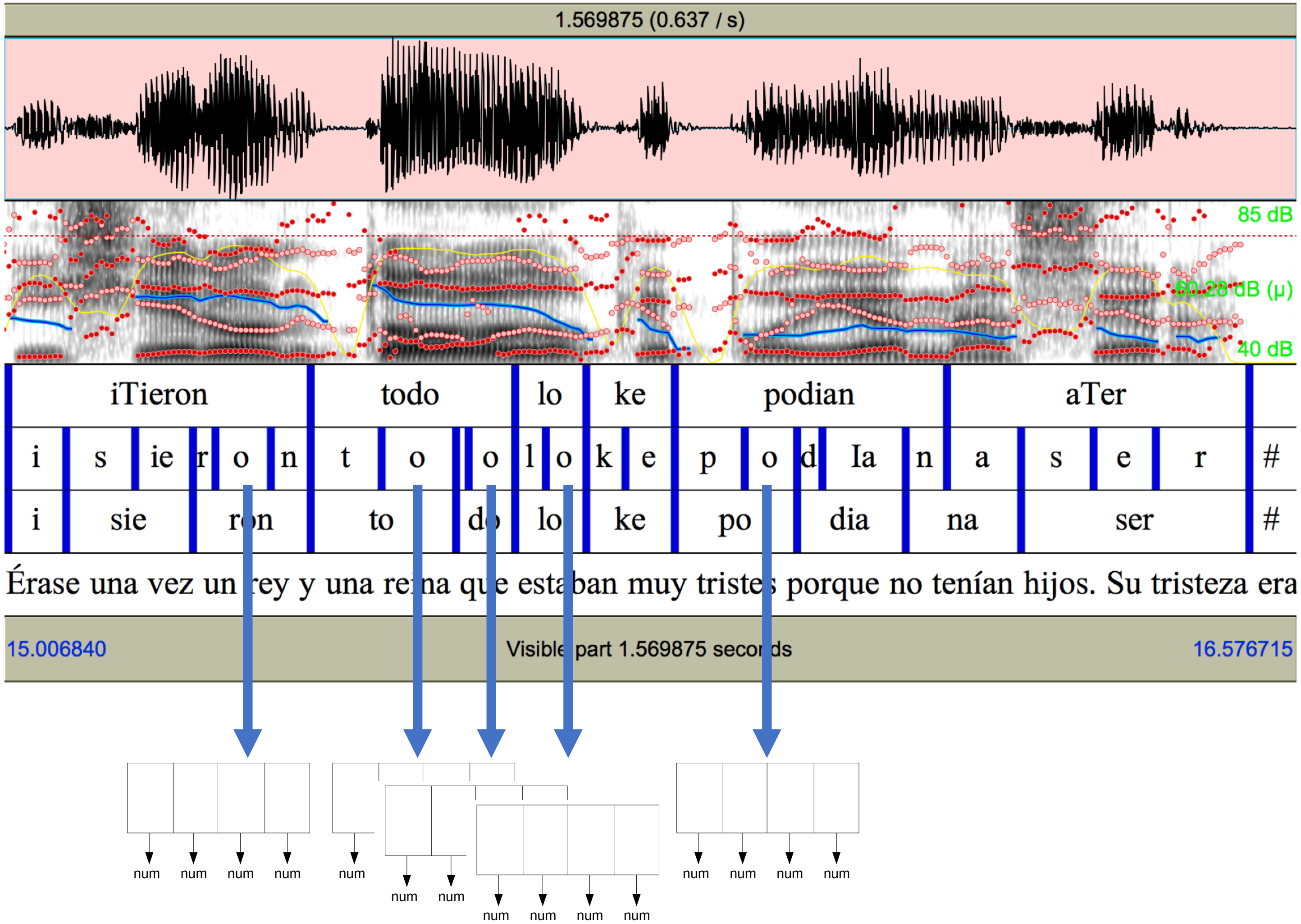
Let’s go through the script for examining acoustic duration together.

1. Download the folder *testing_data* from UB Box. This contains one sound file and an associated textgrid file. The speaker is reading a short text in Spanish.
2. Download and open up *Get_duration_beginners_script.praat*
3. Our first task will be to examine the existing script’s parts.
4. We will do Question 1 together, but then you will do question 2 on your own.

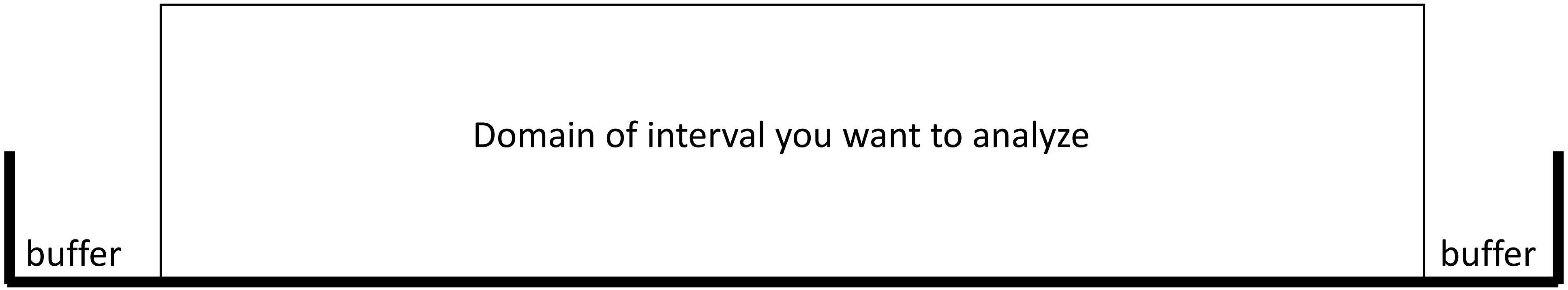
What about dynamic acoustic measures?

- Many acoustic values vary in time. Speech scientists often want to extract a number of measures across a given interval.
- F0, intensity, formants, etc. all change in time because speech is dynamic. The articulators are always moving from one position to the next, rarely staying in the same place for any length of time.
- To capture these movements, we can examine acoustic dynamics – how values change over time.

- To do this, we want to be able to do two things: iterate across a sound file looking for individual intervals to analyze and then, iterate across these intervals to get values at certain points in time.


1.569875 (0.637 / s)
85 dB
60.28 dB (μ)
40 dB
iTieron
todo
lo
ke
podian
aTer
i s ie r o n t o o l o k e p o d Ia n a s e r #
i sie ron to do lo ke po dia na ser #
Érase una vez un rey y una reina que estaban muy tristes porque no tenían hijos. Su tristeza era
15.006840
Visible part 1.569875 seconds
16.576715
num
num
num
num
num
num
num
num
num
num
num
num
num
num
num
num

- In order to track dynamic variables correctly, we need to start our tracking earlier than the temporal domain where we will record values.
- Why? Most tracking algorithms require a “running start” – this is true of autocorrelation/crosscorrelation for f0 tracking, formant tracking, intensity tracking, etc.



Tracking domain

Exercise 1: Examining formants

1. Open up the Praat script *Vowel_Formants_script.praat*
2. We'll be going through this script together, in stages, and talking about the different elements.
3. Three goals here:
   a) Understand the structure of the script and the elements.
   b) Be able to modify parts of the script for other uses (e.g. other acoustic measures)
   c) Be able to run the script to extract formant values from the sound file.

Structure of Vowel_Formants_script.praat

1. Loop across a directory for all sound files.
    a) For each sound file, loop across it to find target vowels
        I. For each vowel, loop across it to cut it up into chunks and extract formant values.

2. Does three things: gets duration of each vowel, gets textual information from the textgrid, and gets a set of formants.