# Praat Scripting for dummies[*]

## Shigeto Kawahara

# 1 Before we begin

## 1.1 Who is this for?

- Those that know basic operations in Praat. See another handout of mine on Praat.

- Those who engage in (phonetic) experiments.

- No background knowledge of programming is assumed.

- If you know how to program, you may find the following discussion too basic.

## 1.2 Why I am writing this handout

- I am willing to admit that I am not born for programming: I was—and am—a computer dummy.

- Hence the disclaimer: my sample scripts that I discuss below may not be perfect. (But who cares as long as they work properly?)

- I know how intimidating programming can be. I really do. I was too scared to write a script for a long time.

- But I am happy that I learned how to script in Praat.

- Hence this handout contains instructions that I wish somebody else had given to me.[1] (The Praat manual does give detailed instructions, so once you know the basics, you can search the Help menu in Praat.)

---

[*]Disclaimer: X for dummies series is published by Wiley-Blackwell. If anybody writes a real Praat Scripting for Dummies for Wiley, I am happy to change the title of this handout. Until then, I cannot think of a better title. This handout is based on a lecture that I gave in the graduate intro phonetics class in Spring 2010. However, this handout itself has not been used for class yet. Any feedback is thus welcome! Revised Dec, 2011

[1]Kathryn Potts Flack (now at Stanford) walked me through one script, which was my very first scripting experience. Thanks!

## 1.3  Why script?

- To save time. Trust me—you can save a lot of time.

- Correct a mistake easily—once you have one script, redoing a similar operation is much easier.

- To be consistent: We can't beat computers in terms of consistency.

- Some operations can be very difficult to do by hand (e.g. get F1 at a mid point of an interval—how do you get "a midpoint" by hand?).

- (Nurture your logical thinking—it does help you to think about which phonetic operations you're applying in what order.)

## 1.4  My personal advice

- "Scripting is about carnivalism" (p.c. anonymous).

- It's ok to steal parts of other people's scripts (in which case you may want to acknowledge).

- You may want to start by modifying other people's scripts.

- Google first! Somebody else may have a script for you already.

# 2  A general idea

- What I do with scripts: process *all* files in a specified folder.

- The basic structure is thus something like

```
start loop here
 do X, Y, Z
end loop here
```

- All you have to do is learn how to X, Y, Z

- And it's easy to learn.

# 3   A first script

- OK, let's look at a sample script now.

- Given below is the one that changes the peak amplitude of all files in a specified folder.

- Open this script (available on my website if you don't have it with you).

- (Save your praat script files with ".praat" extenstion. Tell your computer to remember to open .praat files with Praat (although it really is a text file).)

- Hit run.

- Well, it may have given you an error. But don't get discouraged.

- Now prepare a folder with some sample sound files.

- Prepare an empty folder where you want to save the results.

- Get the paths to these folders, and put them in, when prompted in the input screen.

- Having a trouble? If so, without asking why, go to the scripting window, go Edit and hit Clear History.

- Open one file from the input folder.

- Go back to the script. Hit Paste History.

- It will probably give you a line, something like `Open file...  /x/y/z/w.wav` where `w` is the file name you opened.

- `/x/y/z/` is the path.

```
# This script takes all the files in the specified directory,
# modify their peak to 0.8.
# Place this script in a root folder, which should also contain two subfold
# The input folder should be named inputAmp in which you store the input fi
# The output folder should be named output.

form Files
sentence inputDir ./inputAmp/
sentence outputDir ./output/
endform

# this lists everything in the directory into what's called a Strings list
# and counts how many there are

Create Strings as file list... list 'inputDir$'*.wav
numberOfFiles = Get number of strings

# then it loops through, doing some actions for every file in the list
```

3

```
for ifile to numberOfFiles

# opens each file, one at a time

select Strings list
fileName$ = Get string... ifile
Read from file... 'inputDir$''fileName$'

# and scales peak and write to a new wav file

Scale peak... 0.8
Write to WAV file... 'outputDir$''fileName$'

# then remove all the objects except the strings list so praat isn't all clu

select all
minus Strings list
Remove

endfor

# at the very end, remove any other objects sitting around - like the strin

select all
Remove
```

- Well, this simple script can already be intimidating.

- I was intimidated when I first looked at something like this.

- But let's try. First, you notice that there are many sentences that start with #.

- These sentences are called **comments**.

- They explain (roughly) what the following lines do.

- These are very helpful when you come back to the script and attempt to modify it.

- They are also useful if you're letting other people use your script.

- The first few lines provide a description of the script and how you use it.

- (While you're editing your script, it is advisable that you don't actually delete modified sentences; you comment out so that they are recoverable.)

- Well, anyway, let's remove them, except for one crucial part.

```
form
sentence inputDir ./inputAmp/
sentence outputDir ./output/
endform

Create Strings as file list... list 'inputDir$'*.wav
numberOfFiles = Get number of strings

for ifile to numberOfFiles

select Strings list
fileName$ = Get string... ifile
Read from file... 'inputDir$''fileName$'

# THIS IS WHERE YOU SPECIFY THE OPERATION YOU WANT PRAAT TO DO'

Scale peak... 0.8
Write to WAV file... 'outputDir$''fileName$'

select all
minus Strings list
Remove

endfor

select all
Remove
```

- OK, it's shorter now.

- See where I say "THIS IS WHERE YOU SPECIFY THE OPERATION YOU WANT PRAAT TO DO"?

- This is where you want to specify what you want to do.

- The rest is (mostly) for the loop, so you can actually steal my loop syntax, and modify just that one sentence.

# 4    Modifying your first script

- So how would I know how to modify a command?

- OK, now go to a scripting window.

- Go to Edit, and hit Clear History.

- Do an operation that you want to do.

- Go back to the script that you're editing. Hit Paste History.

- Praat gives you the command.

- Now you know what you did to get the path to an input folder above?

- If this is already enough, all you have to do is replace this command with yours, and you have your own new script!

- An exercise: modify the above script so that it adjusts average amplitude to 70 dB.

- An exercise: modify the above script so that it lengthens all the files by a factor of 1.2.

# 5    Looking at the script in more depth

- Let's look at the script. I explain each portion in the comments.

```
# The following four lines ask a user to specify
# the input folder and the output folder.
# form and endform creates a screen that takes inputs.
# The first argument is a kind of variable (see below)
# The second argument is a variable name (see below)
# The third argument is a default setting
# Variable names in Praat must start with a small letter.

form
sentence inputDir ./inputAmp/
sentence outputDir ./output/
endform

# What Praat does for looping is first to create a so-called string list
# and counts how many files there are in that list (find n).
# This allows us to do operation X for n-times.
# By the way "*" means "everything". So this means "every .wav file"

Create Strings as file list... list 'inputDir$'*.wav
numberOfFiles = Get number of strings
```

```
# "for" is a function for loop.
# "ifile" means as follows:
# start i with 1 and do the operation that follows.
# change i to 2 and do the operation.
# change i to 3....
# keep until i becomes n.

for ifile to numberOfFiles

  # The following three lines open i-th file in the string list
  # Note it says "ifile"

select Strings list
fileName$ = Get string... ifile
Read from file... 'inputDir$''fileName$'

  # THIS IS WHERE YOU SPECIFY THE OPERATION YOU WANT PRAAT TO DO

Scale peak... 0.8

  # Write the output file

 Write to WAV file... 'outputDir$''fileName$'

  # This is for cleaning. You select everything minus the string list.
  # And remove everything from the object window.

  select all
  minus Strings list
  Remove

# This is the end of the loop.

endfor

select all
Remove
```

- Makes sense now? Mostly?

- You're welcome to steal my loop command.

# 6  Variable

- A variable is like your name.

- Before you're born, you don't know how you'd be called.

- But once it's fixed, it's fixed. You will be called with your name.

- In the example above, 'inputDir$' is a variable.

- It's value is unknown until the user specifies it.

- Once they do, it will be the same.

- A string variable in Praat needs to be followed by '$'.

- A numerical variable in Praat should not be followed by '$'.

- When you use a variable in script, you need to put it in quotes.

- Exercise: Modify your above script (change in dB) so that it can take user's input.

- Exercise: Modify your above script (lengthen) so that it can take user's input.

- To define a numerical variable in the input form, start with Positive.

# 7 A loop within a loop: all intervals in all files

- Now you may want to do something to all intervals in all files. (I assume you know about Praat annotation.)

- The next script has a loop within a loop.

- It gets the duration of all intervals in all files.

- The inner loop is a loop for all intervals.

- The outer loop is a loop for all files.

- Let's take a look.

```
# Calculates the duration of all intervals of all the files in a specified
# All you need is a set of TextGrid files.
# Author: Shigeto Kawahara
# Ver. 10/3/2010 (supercedes the previous versions)


form Calculate durations of labeled segments
sentence directory ./
comment The name of the result file
text textfile Result.txt
endform
```

```
#Read all files in a folder
Create Strings as file list... gridlist 'directory$'/*.TextGrid
n = Get number of strings

for i to n
clearinfo

#Loop through files
select Strings gridlist
gridname$ = Get string... i
Read from file... 'directory$'/'gridname$'
soundname$ = selected$ ("TextGrid")

fileappend "'textfile$'" 'soundname$''tab$'

# We then calculate the durations

int=Get number of intervals... 1

for k from 1 to 'int'
select TextGrid 'soundname$'

label$ = Get label of interval... 1 'k'
if label$ <> ""
# calculates the onset and offset
  onset = Get starting point... 1 'k'
   offset = Get end point... 1 'k'

# calculates duration
dur = offset-onset
fileappend "'textfile$'" 'label$''tab$''dur''tab$'
endif
endfor

fileappend "'textfile$'" 'newline$'
endfor

# clean up
select all
Remove
```

# 8   Getting objects

- The final trick.

- To get acoustic properties like formant values, F0 and intensity, you have to get these information as a tier.

- Let's study the following script, which takes a lot of acoustic values.

```
# This script will get various acoustic properties of all intervals of all
# Version: 10 Feb 2010
# Author: Shigeto Kawahara
# Input: TextGrid and wav in the same directly. They must have the same name

form Get acoustic properties of Xitsonga frictives
sentence Directory  ./
comment If you want to analyze all the files, leave this blank
word Base_file_name
comment The name of result file
text textfile Xitsonga.txt
endform

# Write-out the header (copy if necessary)



 fileappend "'textfile$'" soundname'tab$'intervalname'tab$'F2 V1'tab$'F2 of
 fileappend "'textfile$'" 'newline$'

#Read all files in a folder
Create Strings as file list... wavlist 'directory$'/'base_file_name$'*.wav
Create Strings as file list... gridlist 'directory$'/'base_file_name$'*.Text
n = Get number of strings

for i to n
clearinfo

#We first extract a pitch tier, a formant tier, and an intensity tier
select Strings wavlist
filename$ = Get string... i
Read from file... 'directory$'/'filename$'
soundname$ = selected$ ("Sound")
To Formant (burg)... 0 5 5000 0.025 50
select Sound 'soundname$'
To Intensity... 100 0
```

```
# We now read grid files and extract all intervals in them
select Strings gridlist
gridname$ = Get string... i
Read from file... 'directory$'/'gridname$'
int=Get number of intervals... 1

# We then calculate the acoustic properties

for k from 1 to 'int'
select TextGrid 'soundname$'
label$ = Get label of interval... 1 'k'

if label$ ="a"
onsetA = Get starting point... 1 'k'
offsetA = Get end point... 1 'k'
midpointA=(onsetA+offsetA)/2
windowAbegin=midpointA-0.01
windowAend=midpointA+0.01

select Formant 'soundname$'
ftwoA = Get mean... 2 'windowAbegin' 'windowAend' Hertz
fthreeA = Get mean... 3 'windowAbegin' 'windowAend' Hertz
endif

if label$ = "sw" or label$="sh"

# calculates the onset, offset and midpoint
  onset = Get starting point... 1 'k'
   offset = Get end point... 1 'k'
dur = offset-onset

# defining landmarks
preVoffset = onset-0.01
preVonset = onset-0.03

postVonset = offset+0.01
postVoffset = offset+0.03

friconset = onset+0.01
fricoffset = offset-0.01

select Formant 'soundname$'
ftwoOffset = Get mean... 2 'preVonset' 'preVoffset' Hertz
fthreeOffset = Get mean... 3 'preVonset' 'preVoffset' Hertz
ftwoOnset = Get mean... 2 'postVonset' 'postVoffset' Hertz
fthreeOnset = Get mean... 3 'postVonset' 'postVoffset' Hertz
```

```
# calculates intensity
select Intensity 'soundname$'
vowelintensity = Get mean... preVonset preVoffset
intensitymax = Get maximum... onset offset Parabolic
intensityaverage = Get mean... onset offset dB

# COG
select Sound 'soundname$'
Resample... 20000 50
select Sound 'soundname$'_20000
Extract part... 'friconset' 'fricoffset' rectangular 1 no
select Sound 'soundname$'_20000_part
To Spectrum... yes
select Spectrum 'soundname$'_20000_part
cog = Get centre of gravity... 2
sd = Get standard deviation... 2
skew = Get skewness... 2
kurtosis = Get kurtosis... 2

labelline$ ="'soundname$''tab$''label$''tab$'"
fileappend "'textfile$'" 'labelline$'

endif

if label$ ="i"
onsetI = Get starting point... 1 'k'
offsetI = Get end point... 1 'k'
midpointI=(onsetI+offsetI)/2
windowIbegin=midpointI-0.01
windowIend=midpointI+0.01

select Formant 'soundname$'
ftwoPostV = Get mean... 2 'windowIbegin' 'windowIend' Hertz
fthreePostV = Get mean... 3 'windowIbegin' 'windowIend' Hertz
endif
endfor

resultline$ = "'ftwoA''tab$''ftwoOffset''tab$''fthreeA''tab$''fthreeOffset'
fileappend "'textfile$'" 'resultline$'
fileappend "'textfile$'" 'newline$'


endfor

# clean up
```

select all
Remove

# 9 My final advice

- Don't get frustrated. You will figure it out.

- I was a computer dummy, but I could, so you can.

- Relax, breath, but don't give up.

- Again scripting is about stealing (with proper acknowledgements when necessary).