Prosodic analyses and manipulations in Praat

Sandra Muckel Pappert
Universität Leipzig

Graduiertenkolleg
Universalität und Diversität
08. + 11.03.2004

Questions, comments etc are welcome troughout the tutorial and thereafter:

muckels@rz.uni-leipzig.de

Outline

- **Monday 14 - 16**
  - **Praat functionality**
  - **Scripting language**
- **Monday 16 - 18**
  - **A simple script: Segmenting**
- ***Homework***
  - **Your first script: Labelling**
- **Thursday 14 - 16**
  - **Scripting: Reading tables**
  - **Scripting: Pitch**
- **Thursday 16 - 18**
  - **Scripting: Manipulation (pitch & duration)**

http://www.praat.org

Praat: doing Phonetics by Computer - Microsoft Internet Explorer

Datei Bearbeiten Ansicht Favoriten Extras ?

Zurück Suchen Favoriten Medien

Adresse http://www.fon.hum.uva.nl/praat/ Wechseln zu

Links Steffen InfL D3 home FG GK Bahn Google postbank Mensa LEO P&P1 SMWK EvoLang Architecture Karstadt

Praat: doing phonetics by computer

Download Praat:

* Macintosh
* Windows
* Linux, FreeBSD
* SGI, Solaris, HPUX
* the source code

Introductory manuals:

- Choose **Intro** from the **Help** menu in the Praat program.
- Sidney Wood's Praat beginners' manual.
- Pascal van Lieshout's Praat short tutorial.

Paul

The authors

Paul Boersma and David Weenink
Institute of Phonetic Sciences
University of Amsterdam
Herengracht 338
1016CG Amsterdam
The Netherlands

David

Questions, problems, solutions:

1. Many problems can be solved by upgrading to version 4.2 of Praat.
2. Make sure you have read the Intro from Praat's **Help** menu.
3. If that does not help, use the **Search** button in Praat's manual window.
4. Or consult the Frequently Asked Questions directly.
5. There is a user group on the Internet: the Praat User List.
6. If none of the above helps, you may send mail to paul.boersma@uva.nl.

http://www.fon.hum.uva.nl/ Internet

Start server1.rz.uni-l... server1.rz.uni-l... Microsoft Power... Praat: doing P... 19:33

Praat functionality

(3,529 MB)

- Speech analysis:
  - spectrograms
  - pitch analysis
  - formant analysis
  - intensity analysis
  - jitter, shimmer, voice breaks
  - cochleagram
- Labelling and segmentation:
  - Label intervals and time points on multiple tiers
  - use phonetic alphabet
  - use sound files up to 2 GB (3h)
- Graphics:
  - produce eps-files
  - mathematical and phonetic symbols
- Speech synthesis:
  - from pitch, formant, and intensity
  - articulatory synthesis
- Listening experiments:
  - identification and discrimination
- Speech manipulation:
  - change pitch and duration contours
  - filtering
- Statistics:
  - multidimensional scaling
  - principal component analysis
  - discriminant analysis
- Programmability:
  - scripting language
  - communication with other programs

Praat functionality

- Speech analysis:
  - spectrograms
  - pitch analysis
  - formant analysis
  - intensity analysis
  - jitter, shimmer, voice breaks
  - cochleagram
- Labelling and segmentation:
  - Label intervals and time points on multiple tiers
  - use phonetic alphabet
  - use sound files up to 2 GB (3h)
- Graphics:
  - produce eps-files
  - mathematical and phonetic symbols
- Speech synthesis:
  - from pitch, formant, and intensity
  - articulatory synthesis
- Listening experiments:
  - identification and discrimination
- Speech manipulation:
  - change pitch and duration contours
  - filtering
- Statistics:
  - multidimensional scaling
  - principal component analysis
  - discriminant analysis
- Programmability:
  - scripting language
  - communication with other programs

Praat editor windows

- **Praat objects**
- **Praat picture**
- **Manual**
- **Sound**
- **Info**
- **Pitch**
- **Manipulation**
- **...**
- **Script**

- **Scripting language:**
  selecting an editor
  (e.g., for sound blabla.wav)

```
editor Sound blabla
endeditor
```

Scripting language

- **See Manual "Scripting"** praat
- **First steps**
  - **Work with a sample file**
  - **`New praat script; Paste history; Clear history`**
  - **Add script-specific commands, define variables etc.**

Scripting language: Layout

- **White space at the beginning of lines (indenting)**
  can be used to make a skript readable.
- **White space within or at the end of lines**
  will be interpreted, e.g., as an unknown command.
- **Comments**
  are lines that start with `!`, `#` or `;`.
- **There has to be one line per command and one command per line.**
- **Case is distinctive!**
  - **Commands available via the menu begin with letters in upper case.**
  - **Commands available via syntax only begin with letters in lower case.**

Scripting language: Variables

- **Use lower case!**
- **Numeric expressions**
  - **simple definition** `length1 = 1`
  - **definition by operation** `length2 = length1 * 2`
  - **definition by command** `time1 = Get cursor`
- **Strings (dollar and double quotes!)**
  - **predefined** `tab$, newline$`
  - **simple definition** `cond$ = "a"`
    `phrase1$ = "The old lady"`
  - **string functions** `det$ = left (phrase1$, 3)`
    det$ contains the string "The"
- **Variables can be read from a text file (e.g., a cue list).**

Scripting language: Jumps

- **if**

```
#in a Sound editor
clearinfo
cursor = Get cursor
if cursor < 1
    print cursor is below 1
elsif cursor = 1
    print cursor is 'cursor'
else
    print error
endif
```

  - `clearinfo` clears Info editor
  - `print` writes information to Info editor
  - Quotes tell praat to substitute a variable

Scripting language: Loops

- for

```
directory$ = "D:\sandra\praat\"
clearinfo
Create Strings as file list... list 'directory$'\*.wav
number_of_files = Get number of strings
for x from 1 to 'number_of_files'
   select Strings list
   Sort
   current_file$ = Get string... 'x'
   Read from file... 'directory$''current_file$'
   Play
endfor
```

  - Model for working with all (wav-)files in a directory

Scripting language: Loops

- **while**

```
pitch = Get pitch
while pitch = undefined
    Move cursor by...  0.001
    pitch = Get pitch
endwhile
```

  - `undefined` refers to a missing value
  - Time is given in seconds (`0.001` => 1 ms)

Scripting language: Pause the script

- **pause**

```
clearinfo
directory$ = "D:\sandra\praat\"
Create Strings as file list... list 'directory$'\*.wav
number_of_files = Get number of strings
   if number_of_files = 4
       print yes
   else
       print no
   endif
pause Are there 4 sound files?
```

  - **Opens a dialog box with option to continue or to stop**

Scripting language: "Manual" definition of variables

- **form**

```
clearinfo
directory$ = "D:\sandra\praat\"
form Read next sound
    sentence Nextfile .wav
endform
Read from file... 'directory$''nextfile$'
sound$ = nextfile$ - ".wav"
Edit
```

  - **nextfile$ is the string defined via the form**
  - **sound$ is defined as string minus string**

Scripting language: Writing information to a file

- **print and fappendinfo**

```
print error
print 'sound$''tab$''cursor:3''newline$'
```

  - **`print` writes numeric expressions, strings (incl. blanks!), and variables to the Info editor**
  - **Use blanks, `tab$` and `newline$` to format the output**
  - **`newline$` prevents from overwriting**
  - **`cursor:3` is given with 3 decimals**

```
fappendinfo 'directory$'print.out
```

  - **`fappendinfo` appends Info text to an existing (or new) file**
  - **Use `clearinfo` when needed**

Homework: Cue files

- **Write a praat script**
  that helps you to write cues (time points) of several sounds
  to one file.

- **Print one row per sound.**

- **Begin each row with the name of the sound.**

- **Use `form`, `pause`, `Get cursor`, `print`, `fileappend`, `...`**

Homework: Text grids

- **Write a praat script**
  that helps you to write cues (time points) of several sounds
  to one file.

- **Modify cue.scr such that each interval between two cues will be labelled.**

- **Use `To TextGrid ...`**

TextGrid

- A **`TextGrid`** consists of one or several tiers.
- A tier consists of
  - labelled intervals with boundaries **`IntervalTier`**
  - labelled points **`TextTier`**

`Create TextGrid...`
Command in menu **`New`** creates TextGrid without reference to a sound

`To TextGrid... abc ijk xyz ijk`
Command below button **`Annotate-`** creates TextGrid for a specific sound, e.g., considering length

tiers	**`abc ijk xyz`**
interval tiers	**`abc xyz`**
text tier (repeated)	**`ijk`**

Segmenting

- **TextGrid**

```
select Sound 'sound$'
plus TextGrid 'sound$'
Edit
editor TextGrid 'sound$'
endeditor
```

- **Manual segmenting**
  - **Select a tier** => red
  - **Select a position or interval in the spectogram** => yellow
  - **Add boundaries and/or intervals on selected tier** => blue

`Add interval on tier 1`

- **command in menu `Interval` creates one boundary (at the left) (?)**

`Add on selected tier` or <enter>

- **command in the menu `Boundary` creates one boundary also**

Labelling and writing to a file

- **Labelling**
  - **Just type labels into the text field in the TextGrid editor**
  - **Or select TextGrid in object list and use command below button `Modify-`**

    `Set interval text... 'tier_no' 'interval_no' 'bla'`
- **Writing to a file**
  - **in TextGrid editor** **`Write TextGrid to text file...`**
  - **in Praat objects** **`Write to text file...`**

    `Write to short text file...`

    `Write to chronological text file...`
  - **see text file examples**
  - **bibiko.scr**
  - **textgrid.scr**
- **Sound and TextGrid are stored as separate files!**

Read variables from text files

```
Read Table from table file... 'directory$'cue.out
#first row gives column labels
columns = Get number of columns
   for a from 1 to columns
        Get column label... a
   endfor
rows = Get number of rows
   for b from 1 to rows
        subj'b' = Get value... b subj
        item'b' = Get value... b item
        cond'b' = Get value... b cond
   endfor
clearinfo
```

- **Use cue-data as input for pitch analysis.**
- **Sort rows as file names will be sorted!**
- **Alternative? Try `Select rows where column...`**

Pitch settings

- **In the Sound or TextGrid editor**
  `Show pitch` **(if pitch is not shown)** show_pitch.scr
  `Time step settings... Fixed 0.001 100`
  `Pitch settings... 75 500 Hertz`
  - **Suggested for a male:** `75 300 Hertz`
  - **Suggested for a female:** `100 600 Hertz`
- **In Praat objects, a separate Pitch object is created**
  `To pitch... 0.001 75 500`
- **If you set the floor of the pitch range too low, you will miss very fast F0 changes (analysis window = 3/pitch floor).**
  - settings.scr

Pitch analysis

- **Continuos data**
  - pitchtier.scr
- **In the Sound or TextGrid editor**

```
Move cursor to... 0.5
Get pitch
Select... 0.024 0.37
Get maximum pitch
Move cursor to maximum pitch
Get cursor
```

- **In Praat objects, with a Pitch object selected**

```
Get value at time... 0.5
Get maximum... 0.024 0.37 Hertz Parabolic
Get time of maximum... 0.024 0.37 Hertz Parabolic
Get mean... 0.7 1.031 Hertz
```

  - pitch_div.scr

Manipulation

- **Select a sound in Praat objects and chose `To Manipulation...`**
- **The newly created Manipulation object contains**
  - **the original sound**
  - **a PointProcess representing glottal pulses**
  - **a PitchTier**
  - **a DurationTier**
- **Manipulate pitch and duration of the Manipulation object (either in Praat objects or) in the Manipulation editor.**

- **Intensity is modified separately. Chose `To Intensity...`, `Down to IntensityTier`, select Sound & IntensityTier, `Edit`**

Manipulation of Pitch

- **Manually**
  - **Reduce number of pitch points**
    ```
    Select... 0 3.2
    Stylize Pitch (2 st)
    ```
  - **Optionally**
    ```
    Set pitch dragging strategy... Single all,
    multiple only vertical
    ```
  - **Then drag pitch points up and down, left and right.**
- **Automatized**
  ```
  Select... 0 3.2
  Shift pitch frequencies... -20 Hertz
  ```
- **"All-out" solution** mod_pitch.scr
  ```
  Remove pitch point(s)
  Add pitch point at... 'time' 'f0-value'
  ```

Manipulation of Duration

- **The DurationTier is initially empty.**
  A base line (= 1) marks the original length.
- **Add duration points by clicking on the DurationTier.**
  - **click below the baseline => shorten interval**
  - **click above the baseline => lengthen interval**
- **Or add accurate duration points.**

  `Move cursor to... 2.5`

  `Add duration point at cursor`
  - **duration point will be added at baseline (= 1)**

  `Add duration point at 'time' 'multiplier'`
  - **multiply, e.g., by target length/original length**
    mod_duration.scr

Create Sound from Manipulation

- **In Manipulation editor**

```
Publish resynthesis
endeditor
```

- **In Praat objects**

```
Select Manipulation 'sound$'
Get resynthesis (PSOLA)
```

  - **New pitch points are generated.**
  - **All new pulses within voiceless intervals are removed.**
  - **Voiceless parts are inserted from the original Sound.**

```
Write to WAV file... mod'sound$'.wav
```