

ProsodyPro—A Tool for Large-scale Systematic Prosody Analysis

Yi Xu

University College London

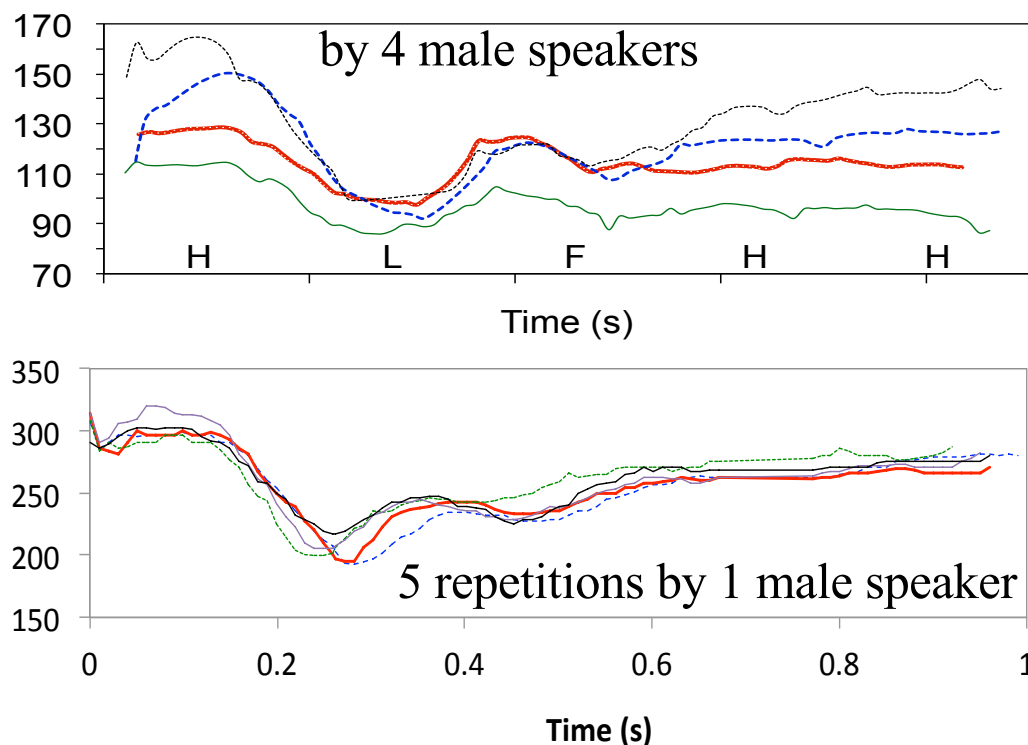
Why large-scale and systematic?

- The key to an accurate experimental observation is that data are as free of *noise* and *confounds* as possible
 - ✧ Strategy 1: Having *multiple speakers* produce the same patterns with *multiple repetitions*
 - ✧ Strategy 2: Making comparisons *as directly as possible*
- But this also means a lot of raw data

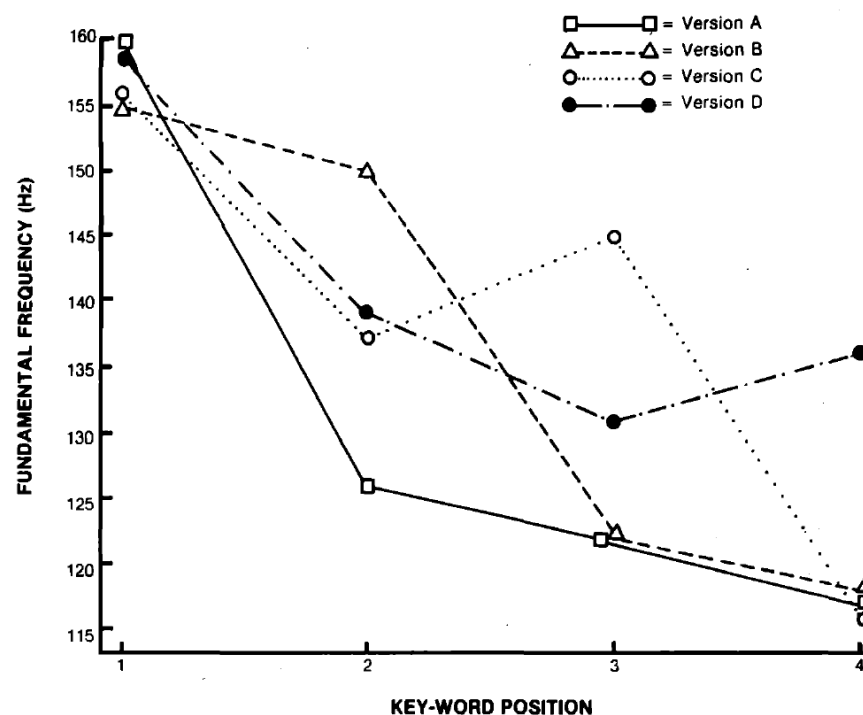
Dilemma in handling a lot of data

- To see all the details,
- or to systematically compare only a few points?

Raw data from Xu (1999)



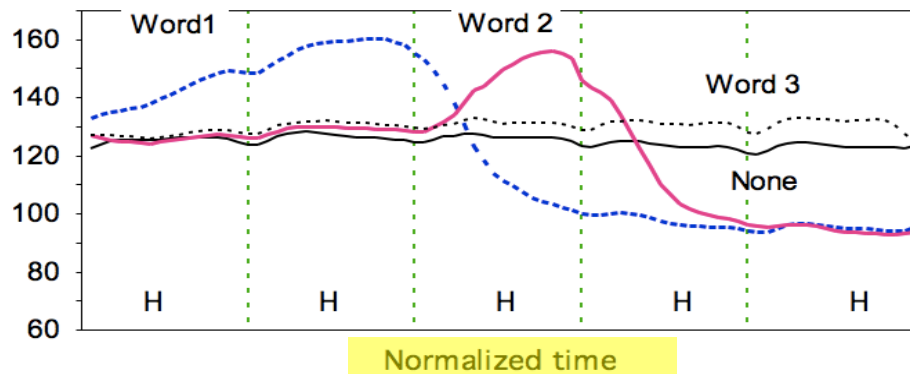
Cooper, Eady & Mueller (1985)



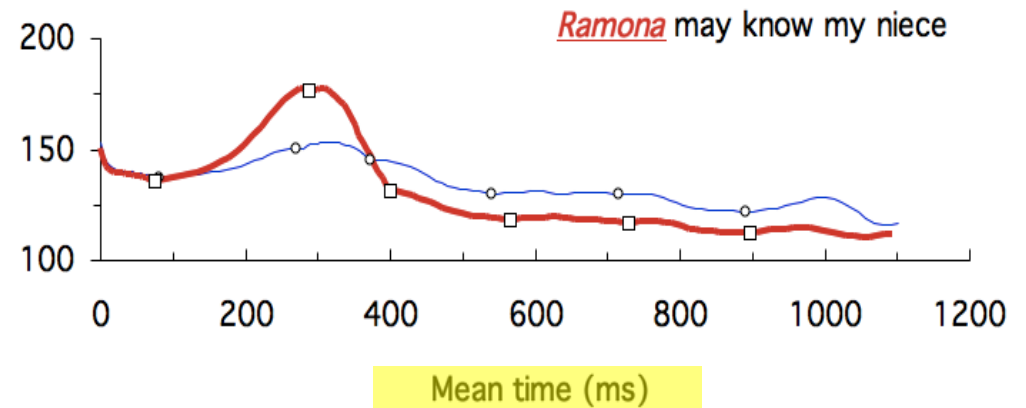
The *fish* will be *fresh* and *cheap* at this *restaurant*.

- *Is it possible to do both?*

Yes, it is possible to make comparisons of large amount of prosody data in fine detail



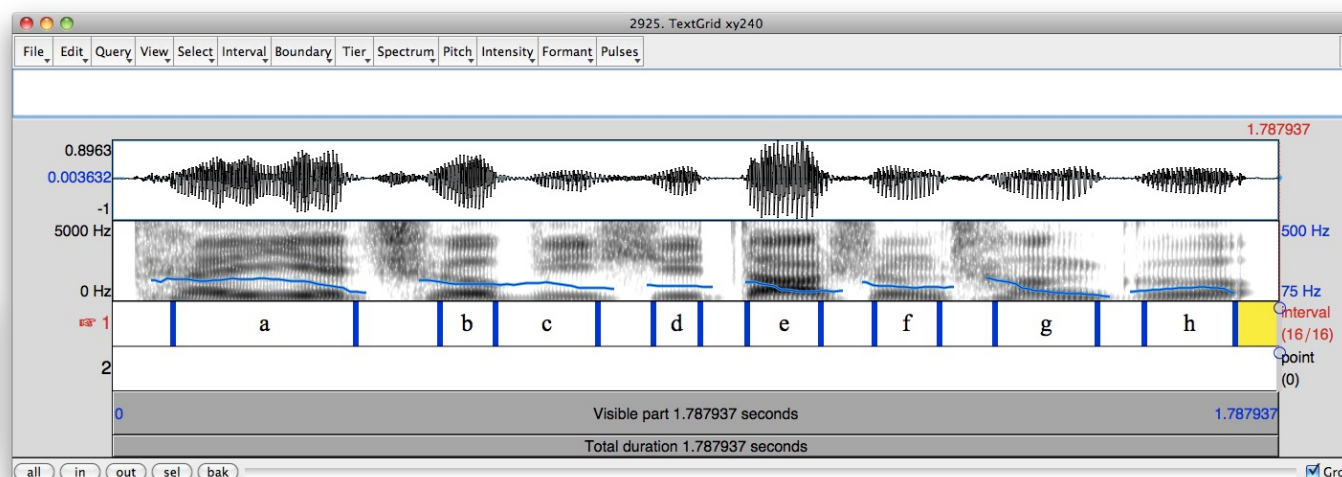
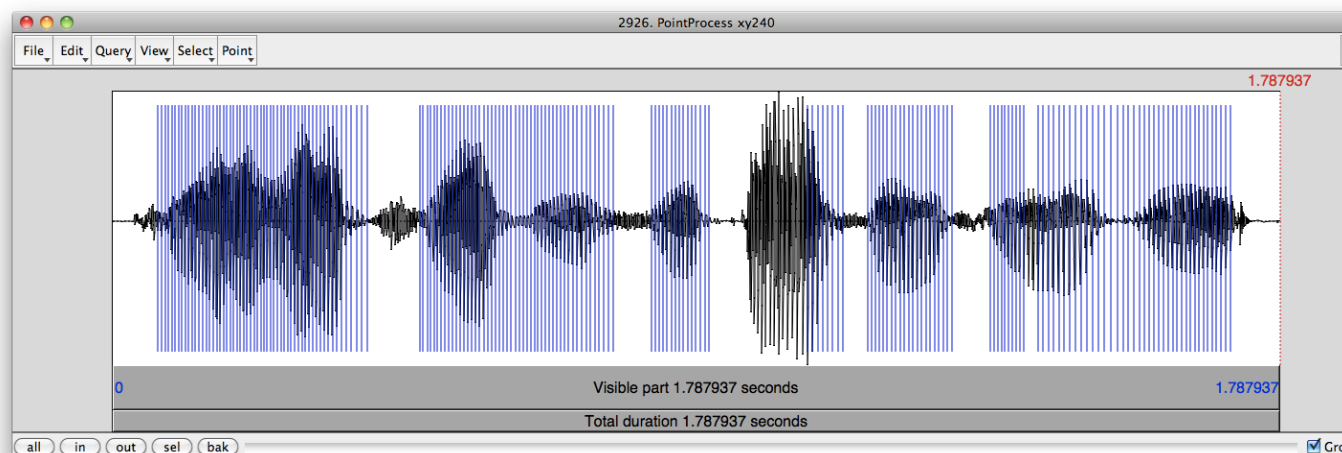
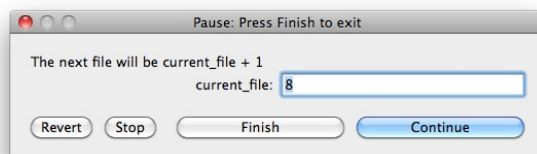
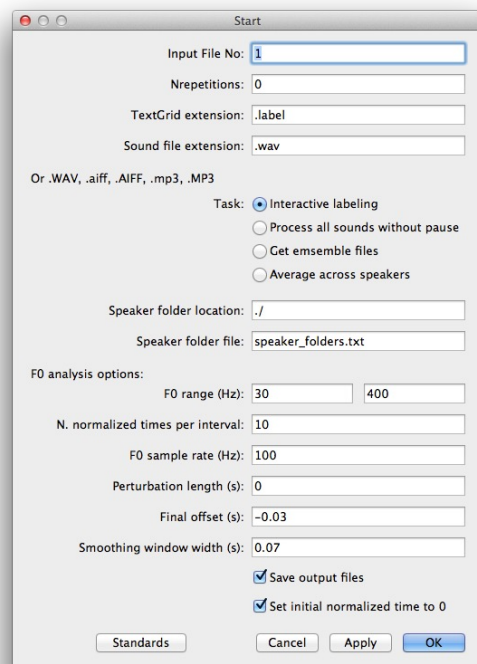
Average of 20 repetitions by 4 male subjects (Xu 1999)



Average of 49 repetitions by 7 subjects (Xu & Xu 2005)

- ✦ In both cases, the effects of focus, tone and stress on every part of the sentence are laid bare
- ✦ This is not only because all the other factors are kept constant, but also because fully continuous F₀ contours are overlaid on top of each other, making the comparisons as direct as possible

ProsodyPro — A Praat script that facilitates large-scale systematic prosody analysis (www.phon.ucl.ac.uk/home/yi/ProsodyPro/)



ProsodyPro — A Praat script that facilitates large-scale systematic prosody analysis (www.phon.ucl.ac.uk/home/yi/ProsodyPro/)

- *Enables direct and systematic comparison of continuous F_0 contours*
- Minimizes human labor by automating tasks that do not require judgment, such as locating and opening sound files, taking measurements, and saving raw results in formats ready for further analysis.
- Enables human intervention of processes that are prone to error by automatic algorithms: pitch detection and segmentation.

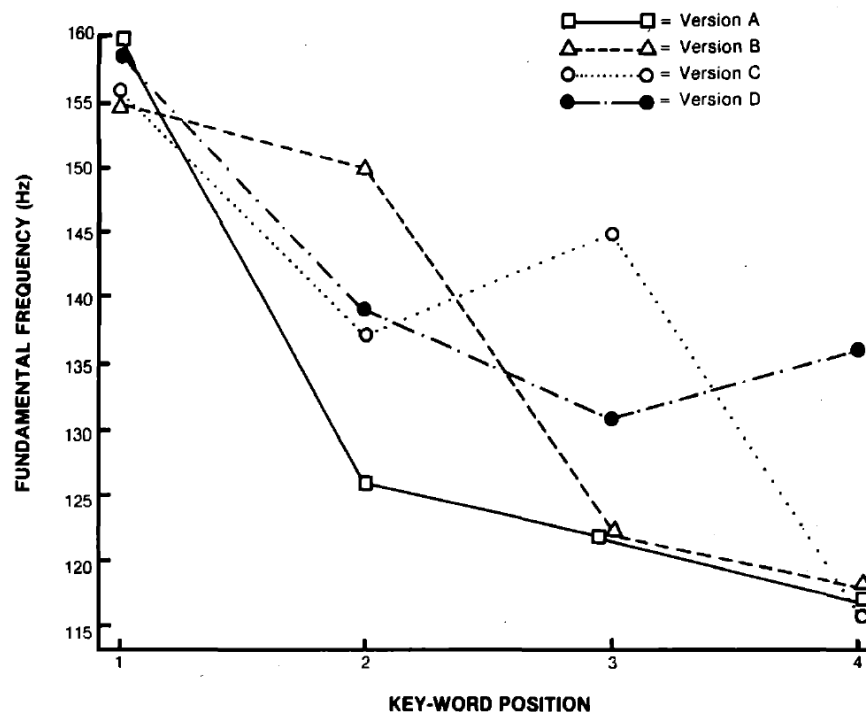
ProsodyPro enables you to:

- ◆ **Get manually rectified F_0 tracks**
- ◆ Segment and label intervals for each sound (.wav) file
- ◆ **Cycle through all sound files in a folder without using menu commands**
- ◆ **Get time-normalized F_0 , F_0 velocity and intensity contours**
- ◆ Get time-normalized F_0 , F_0 velocity and intensity with original time preserved
- ◆ Get rectified, trimmed F_0 as PitchTier objects which can replace the pitch tier in Manipulation objects
- ◆ Get sampled F_0 at fixed time intervals
- ◆ Get continuous F_0 velocity (= first derivative of F_0) curve
- ◆ Get maxf0, minf0, f0range(st), meanf0, mean intensity, duration, max velocity, final velocity, final F_0 and meanintensity from each labeled interval
- ◆ **Get results in ensemble files:** normf0.txt, normIntensity, samplef0.txt, f0velocity.txt, maxf0.txt, minf0.txt, f0range.txt meanf0.txt, maxvelocity.txt, duration.txt, finalvelocity.txt, finalf0.txt and meanintensity.txt
- ◆ **Get mean_normf0 averaged across repetitions of identical sentences**
- ◆ **Get mean_normf0 averaged across multiple speakers**

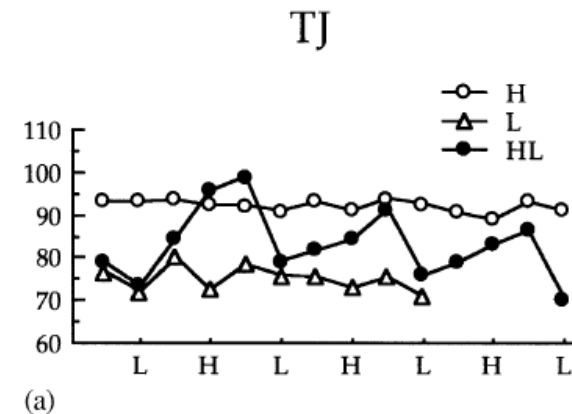
What is time normalization?

- Time-normalization is simply taking the same number of measurements from each interval of interest at even distance,
- which is not unlike methods that take one or two measurements from each interval

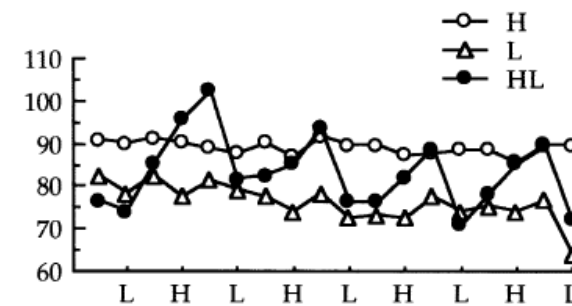
The *fish* will be *fresh* and *cheap* at this *restaurant*.



Cooper, Eady & Mueller (1985)



(a)

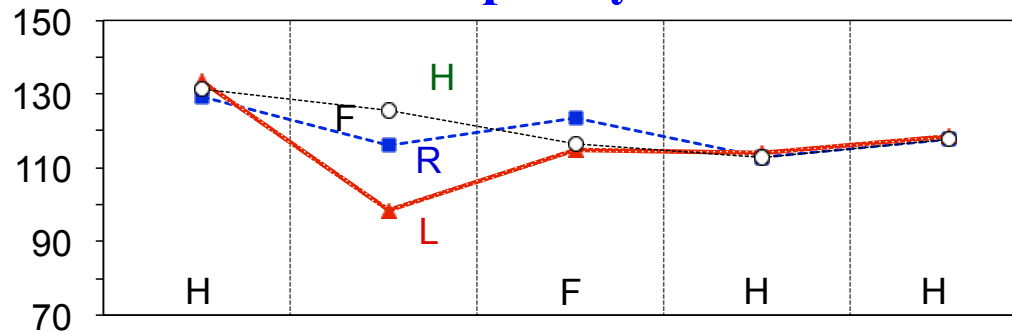


(b)

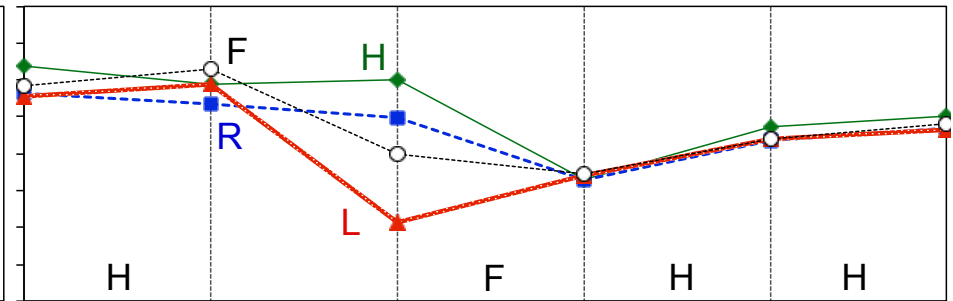
Laniran & Clements (2003)

But why more measurements?

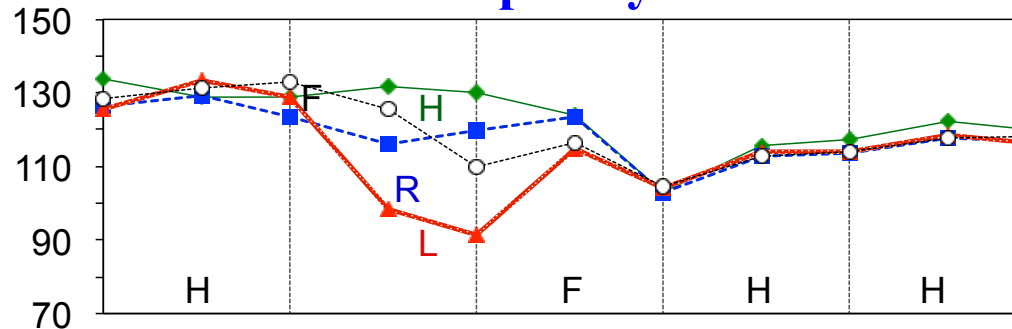
1 samples/syllable



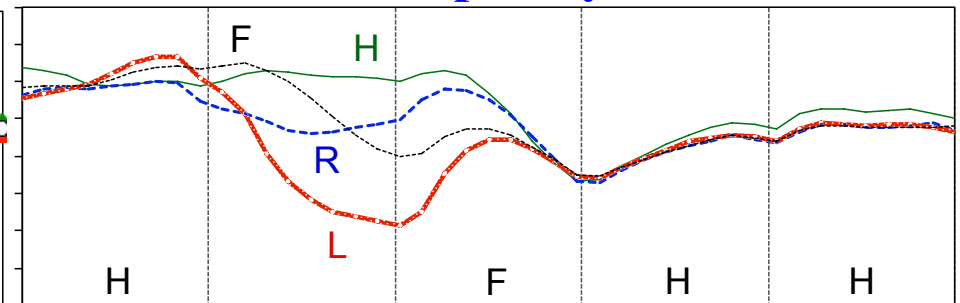
2 samples/syllable



3 samples/syllable



8 samples/syllable



- With 1 measurement from each interval, most details are missing
- With 2-3 measurements per interval, more details start to emerge
- With 8 measurements per interval, fine details of the continuous F_0 contours can be clearly seen

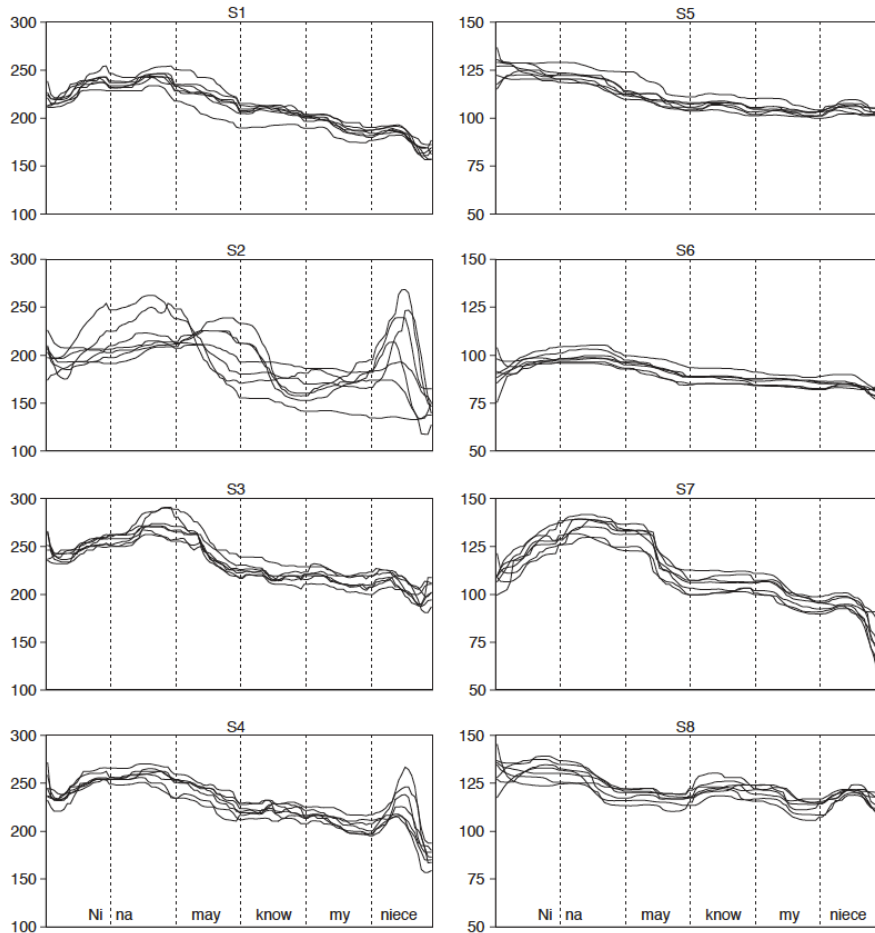
Time normalization

- ✧ Thus time-normalization is only a further extension of the coarser sampling, which is usually not viewed as time-normalization.
- ✧ But the finer sampling in time-normalization shows much more *details*, leaving little to guesswork.
- ✧ More importantly, time-normalization allows averaging across repetitions as well as speakers. This neutralizes unintended random variations and speaker-specific, as opposed to language-specific, features.
- ✧ Also importantly, time-normalization does not mean loss of timing information, as the time value at each measurement point can be recorded, as is done in ProsodyPro.
- ✧ Finally, measurements for statistical analysis are not taken from time-normalized contours in ProsodyPro, but from the raw data.

Individual differences and timing details can still be examined with time-normalization

F_0 contours of one sentence by 8 speakers

Mean F_0 contours with averaged real time



Normalized time

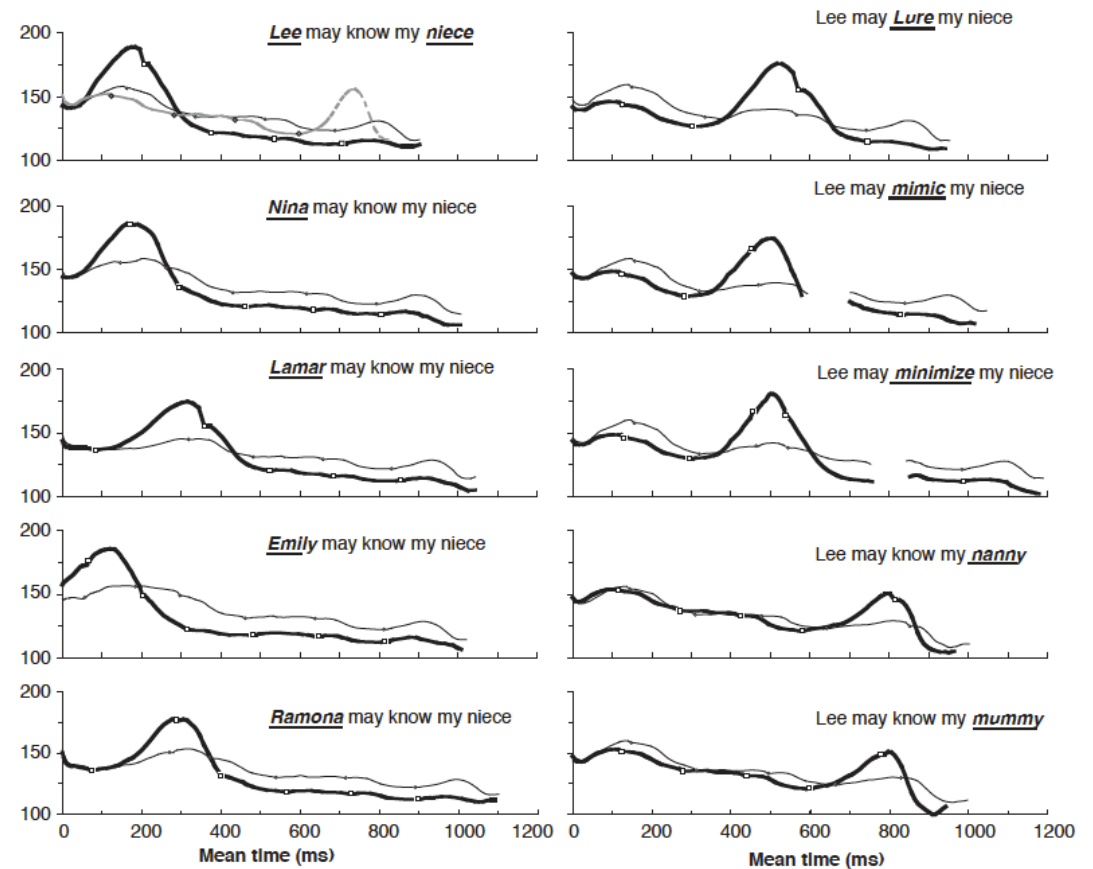
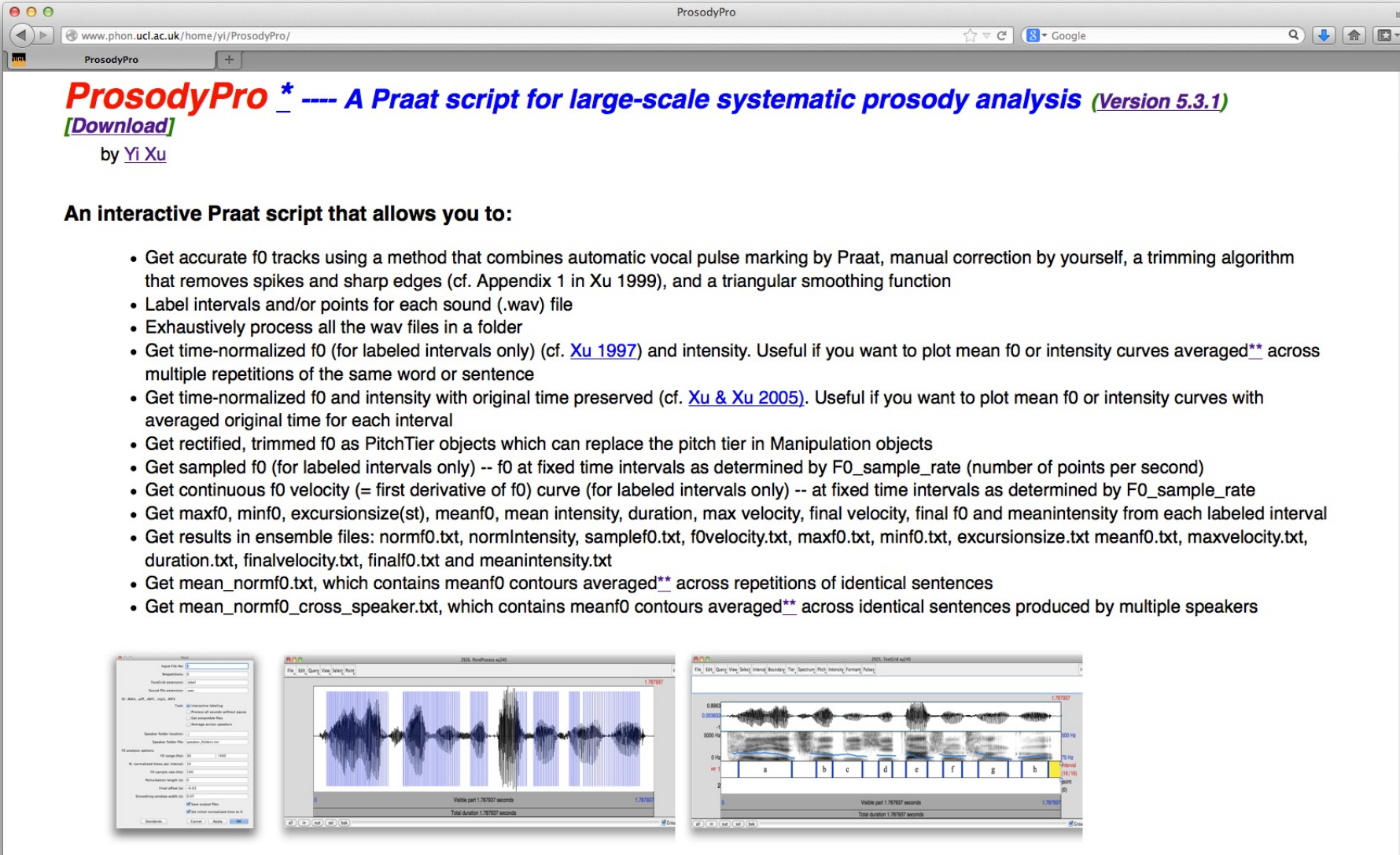


Fig. 3 and 4 from Xu & Xu (2005)

Download ProsodyPro.praat from:

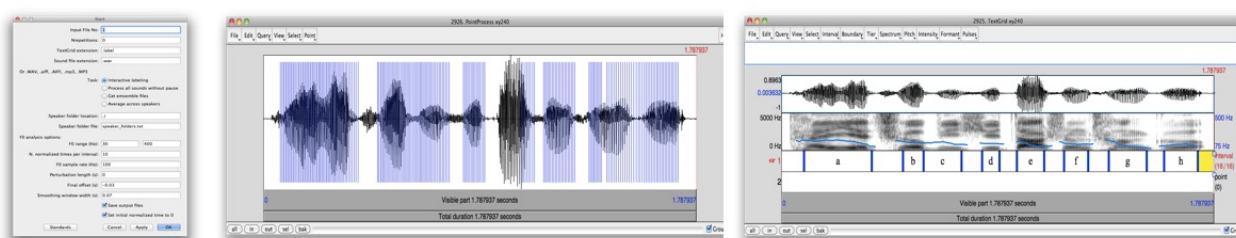
www.phon.ucl.ac.uk/home/yi/ProsodyPro/



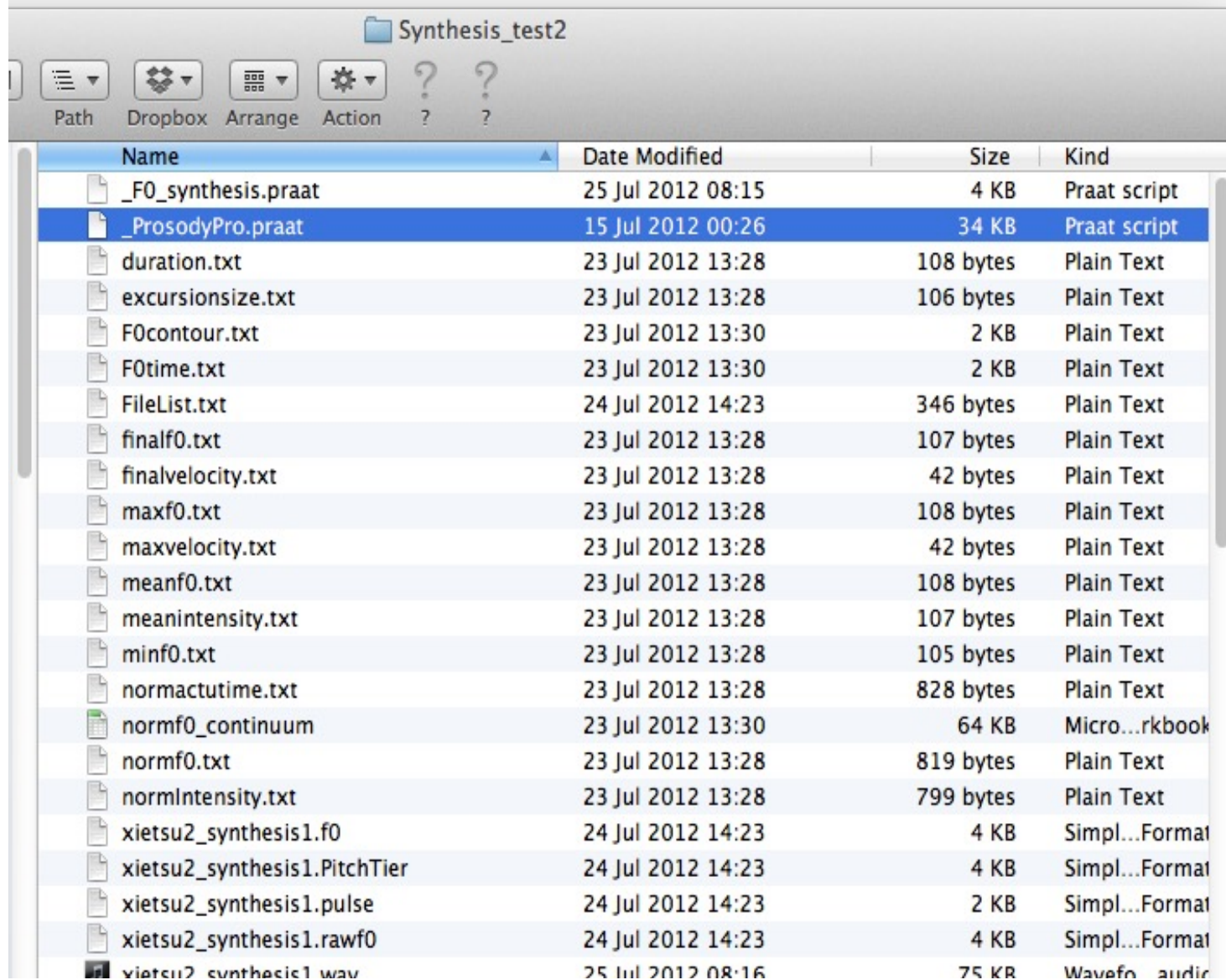
ProsodyPro * --- A Praat script for large-scale systematic prosody analysis (Version 5.3.1)
[Download]
by Yi Xu

An interactive Praat script that allows you to:

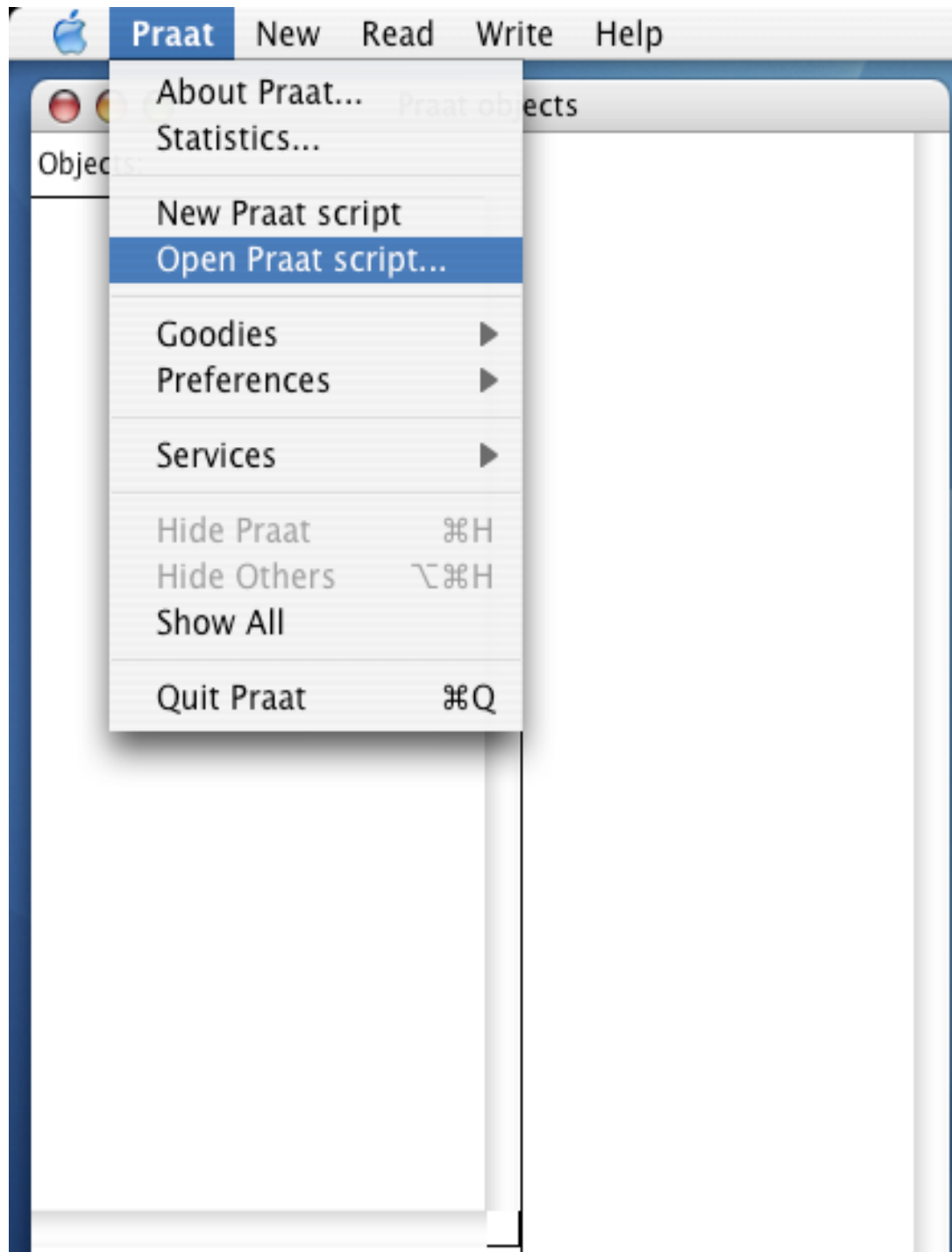
- Get accurate f0 tracks using a method that combines automatic vocal pulse marking by Praat, manual correction by yourself, a trimming algorithm that removes spikes and sharp edges (cf. Appendix 1 in Xu 1999), and a triangular smoothing function
- Label intervals and/or points for each sound (.wav) file
- Exhaustively process all the wav files in a folder
- Get time-normalized f0 (for labeled intervals only) (cf. Xu 1997) and intensity. Useful if you want to plot mean f0 or intensity curves averaged** across multiple repetitions of the same word or sentence
- Get time-normalized f0 and intensity with original time preserved (cf. Xu & Xu 2005). Useful if you want to plot mean f0 or intensity curves with averaged original time for each interval
- Get rectified, trimmed f0 as PitchTier objects which can replace the pitch tier in Manipulation objects
- Get sampled f0 (for labeled intervals only) -- f0 at fixed time intervals as determined by F0_sample_rate (number of points per second)
- Get continuous f0 velocity (= first derivative of f0) curve (for labeled intervals only) -- at fixed time intervals as determined by F0_sample_rate
- Get maxf0, minf0, excursionsize(st), meanf0, mean intensity, duration, max velocity, final velocity, final f0 and meanintensity from each labeled interval
- Get results in ensemble files: normf0.txt, normIntensity, samplef0.txt, f0velocity.txt, maxf0.txt, minf0.txt, excursionsize.txt meanf0.txt, maxvelocity.txt, duration.txt, finalvelocity.txt, finalf0.txt and meanintensity.txt
- Get mean_normf0.txt, which contains meanf0 contours averaged** across repetitions of identical sentences
- Get mean_normf0_cross_speaker.txt, which contains meanf0 contours averaged** across identical sentences produced by multiple speakers



Put ProsodyPro.praat in a folder containing .wav files



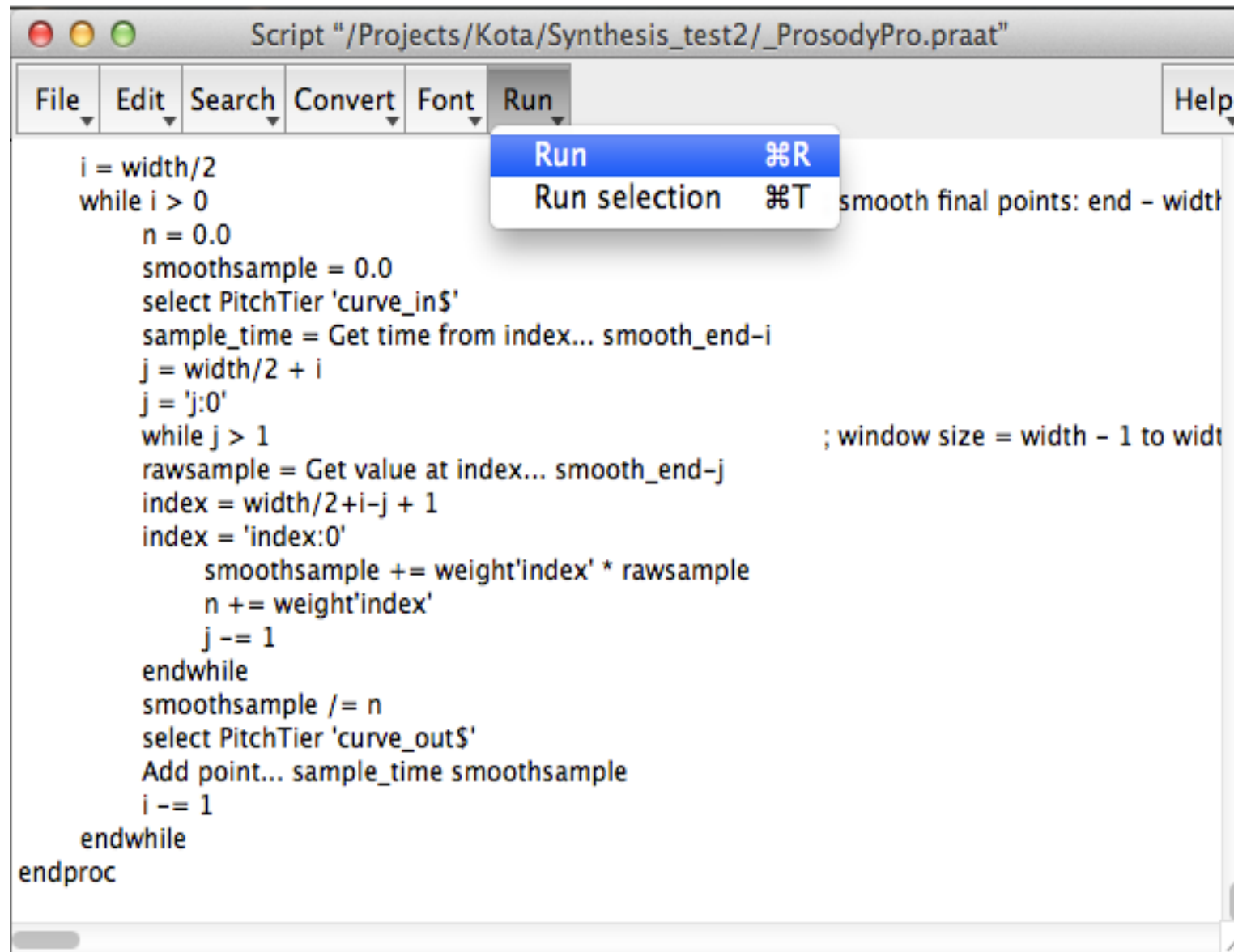
Name	Date Modified	Size	Kind
_F0_synthesis.praat	25 Jul 2012 08:15	4 KB	Praat script
_ProsodyPro.praat	15 Jul 2012 00:26	34 KB	Praat script
duration.txt	23 Jul 2012 13:28	108 bytes	Plain Text
excursionsize.txt	23 Jul 2012 13:28	106 bytes	Plain Text
F0contour.txt	23 Jul 2012 13:30	2 KB	Plain Text
F0time.txt	23 Jul 2012 13:30	2 KB	Plain Text
FileList.txt	24 Jul 2012 14:23	346 bytes	Plain Text
finalf0.txt	23 Jul 2012 13:28	107 bytes	Plain Text
finalvelocity.txt	23 Jul 2012 13:28	42 bytes	Plain Text
maxf0.txt	23 Jul 2012 13:28	108 bytes	Plain Text
maxvelocity.txt	23 Jul 2012 13:28	42 bytes	Plain Text
meanf0.txt	23 Jul 2012 13:28	108 bytes	Plain Text
meanintensity.txt	23 Jul 2012 13:28	107 bytes	Plain Text
minf0.txt	23 Jul 2012 13:28	105 bytes	Plain Text
normactutime.txt	23 Jul 2012 13:28	828 bytes	Plain Text
normf0_continuum	23 Jul 2012 13:30	64 KB	Micro...rkbook
normf0.txt	23 Jul 2012 13:28	819 bytes	Plain Text
normIntensity.txt	23 Jul 2012 13:28	799 bytes	Plain Text
xietsu2_synthesis1.f0	24 Jul 2012 14:23	4 KB	Simpl...Format
xietsu2_synthesis1.PitchTier	24 Jul 2012 14:23	4 KB	Simpl...Format
xietsu2_synthesis1.pulse	24 Jul 2012 14:23	2 KB	Simpl...Format
xietsu2_synthesis1.rawf0	24 Jul 2012 14:23	4 KB	Simpl...Format
xietsu2_synthesis1.wav	25 Jul 2012 08:16	75 KB	Wavefo... audio



**Open ProsodyPro
from within Praat**

**Or launch it from
the finder**

Execute ProsodyPro by choosing “Run” from the Run menu in the Script window (or Control-R)



Start

Input File No:

Nrepetitions:

TextGrid extension:

Sound file extension:

Or .WAV, .aiff, .AIFF, .mp3, .MP3

Task: ☒ Interactive labeling
☐ Process all sounds without pause
☐ Get emsemble files
☐ Average across speakers

Speaker folder location:

Speaker folder file:

F0 analysis options:

F0 range (Hz):

N. normalized times per interval:

F0 sample rate (Hz):

Perturbation length (s):

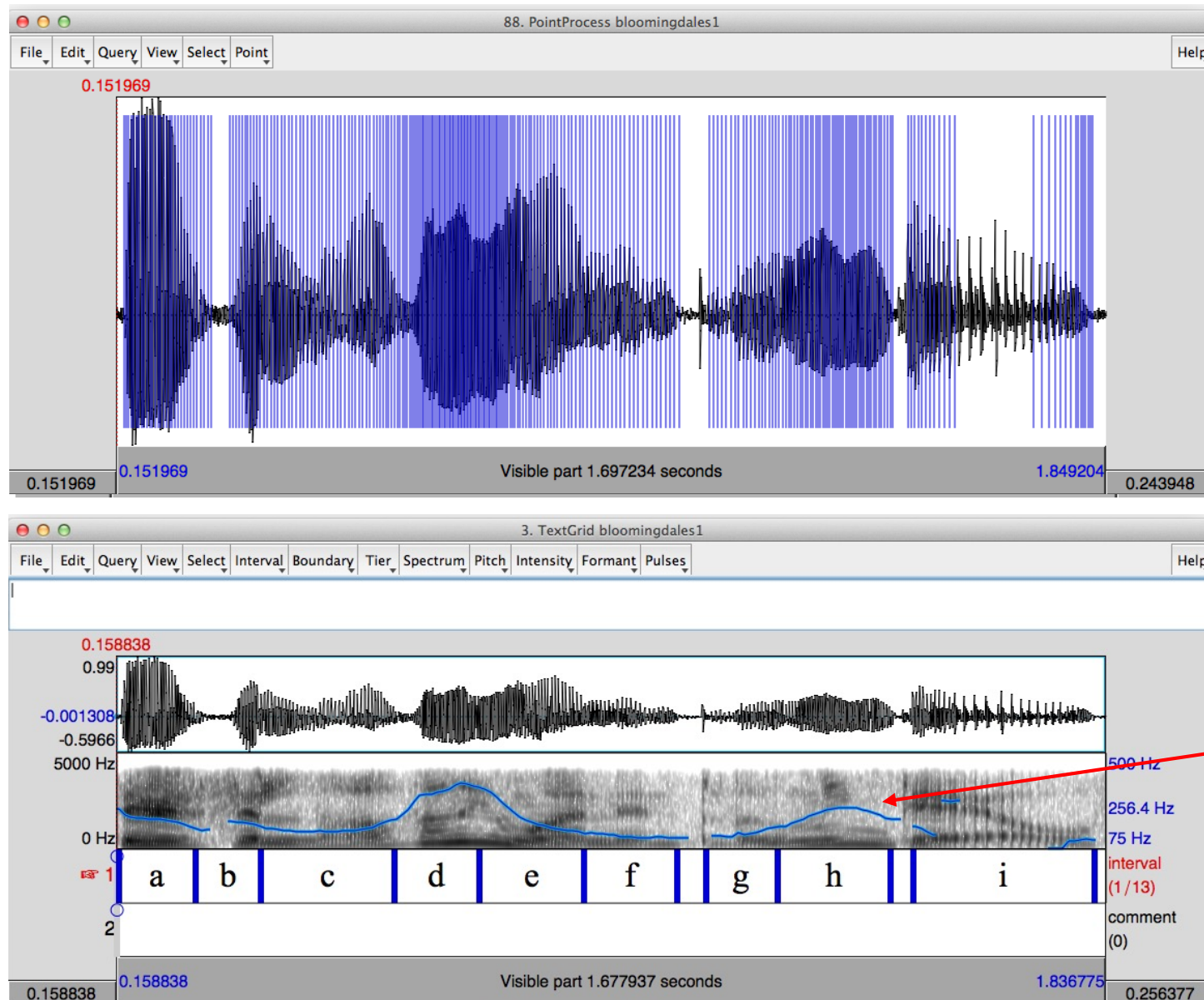
Final offset (s):

Smoothing window width (s):

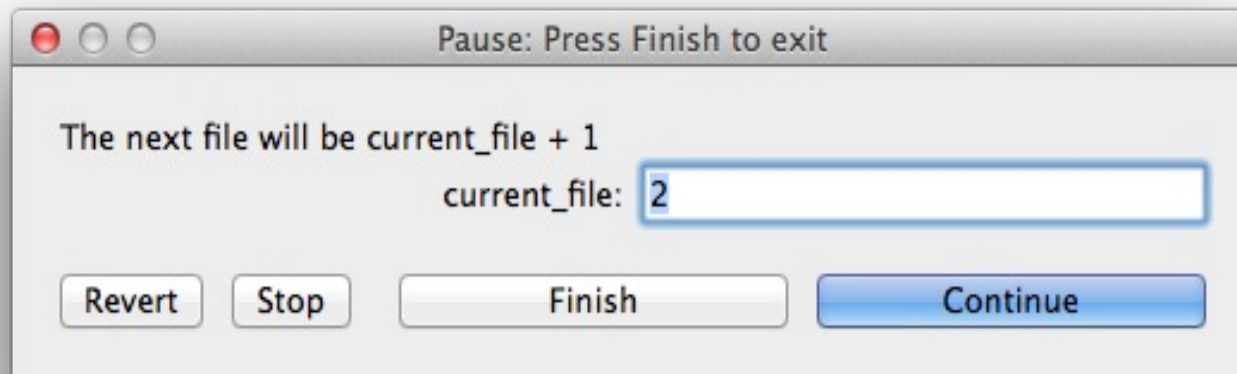
☒ Save output files
☒ Set initial normalized time to 0

In the “Start” window,
check or uncheck the
functions to be
activated

Marking vocal periods (top) and labeling analysis intervals (bottom)



The “Pause” window is for controlling workflow during the execution of ProsodyPro



Continue — Save current data and proceed to the next file

Finish — Save current data and end the execution

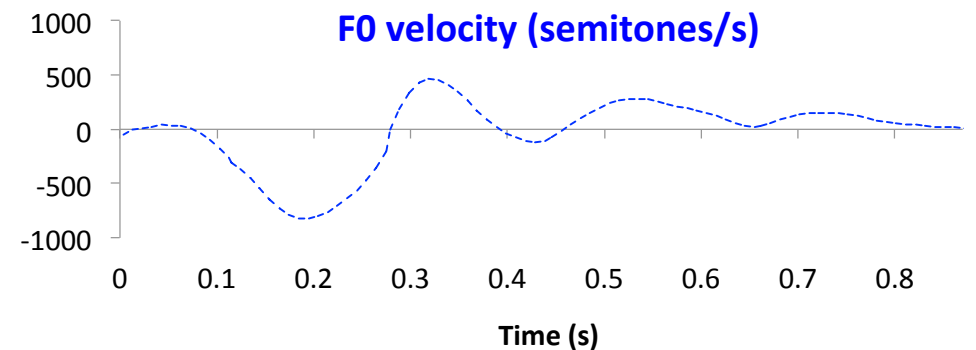
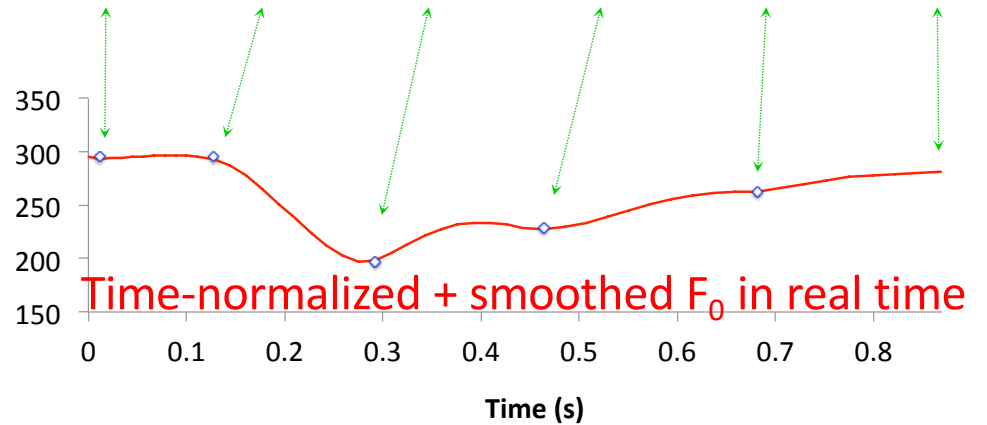
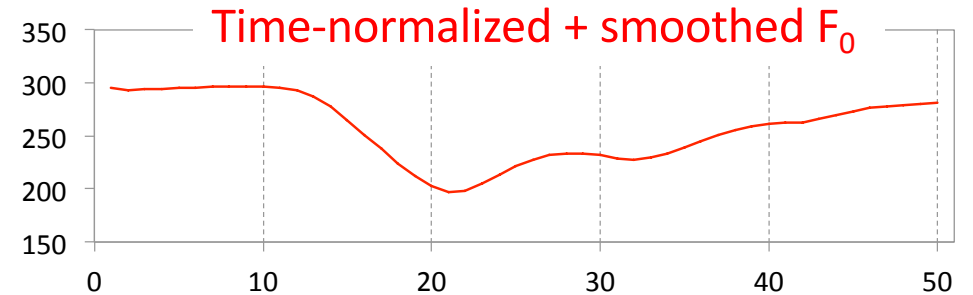
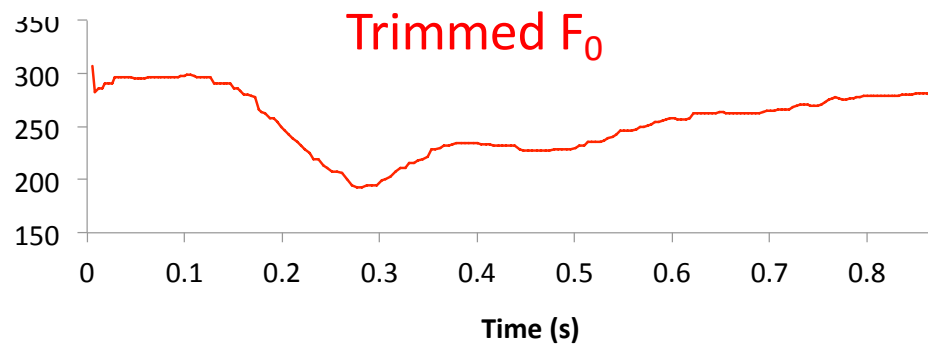
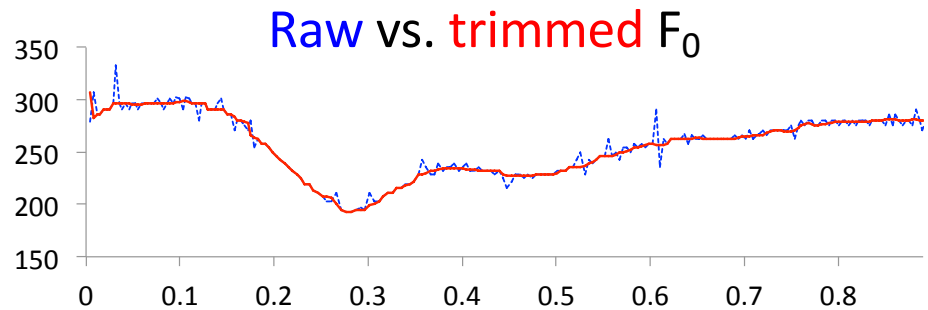
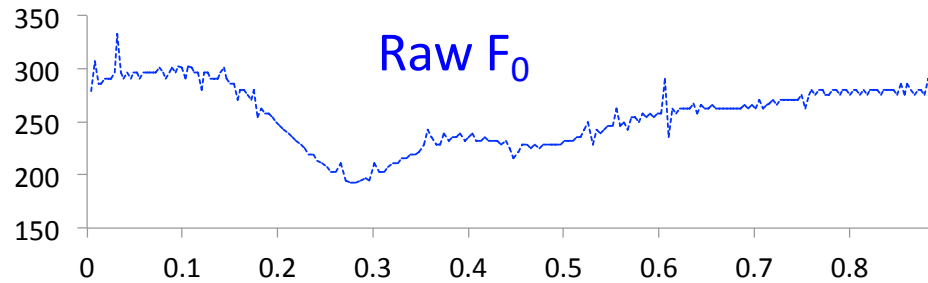
Stop — End the execution without saving the current data

Revert — Revert current file to the initial number

Output files generated for each sound

1. **.rawf0** — F_0 values as inverse of distances between adjacent vocal pulses.
2. **.f0** — F_0 smoothed with a trimming algorithm (Xu, 1999)
3. **.normtimef0** — Time-normalized F_0 . The F_0 in each interval is divided into the same number of points (default = 10)
4. **.actutimenormf0** — Time-normalized F_0 with real time scale as X-axis.
* The time-normalized F_0 files are generated *only for labeled intervals*.
5. **.means** — mean F_0 , mean intensity, duration and peak velocity (if applicable) of all *labeled intervals*.
6. **normf0.txt, samplef0.txt, maxvelocity.txt, meanintensity.txt, duration.txt, meanf0.txt** — Ensemble files containing output data for individual .wav files put together.
7. **mean_normf0_cross_speaker.txt** — Grand ensemble file containing mean norm-time F_0 averaged across speakers.

Some curves generated by ProsodyPro



$$F_0' = (F_{0i+1} - F_{0i-1}) / (t_{i+1} - t_{i-1})$$

Start

Input File No: 1

Nrepetitions: 5

TextGrid extension: .label

Sound file extension: .wav

Or .WAV, .aiff, .AIFF, .mp3, .MP3

Task: ☐ Interactive labeling
☐ Process all sounds without pause
☒ Get emsemble files
☐ Average across speakers

Speaker folder location: ./

Speaker folder file: speaker_folders.txt

F0 analysis options:

F0 range (Hz): 30 400

N. normalized times per interval: 10

F0 sample rate (Hz): 100

Perturbation length (s): 0

Final offset (s): -0.03

Smoothing window width (s): 0.07

☒ Save output files

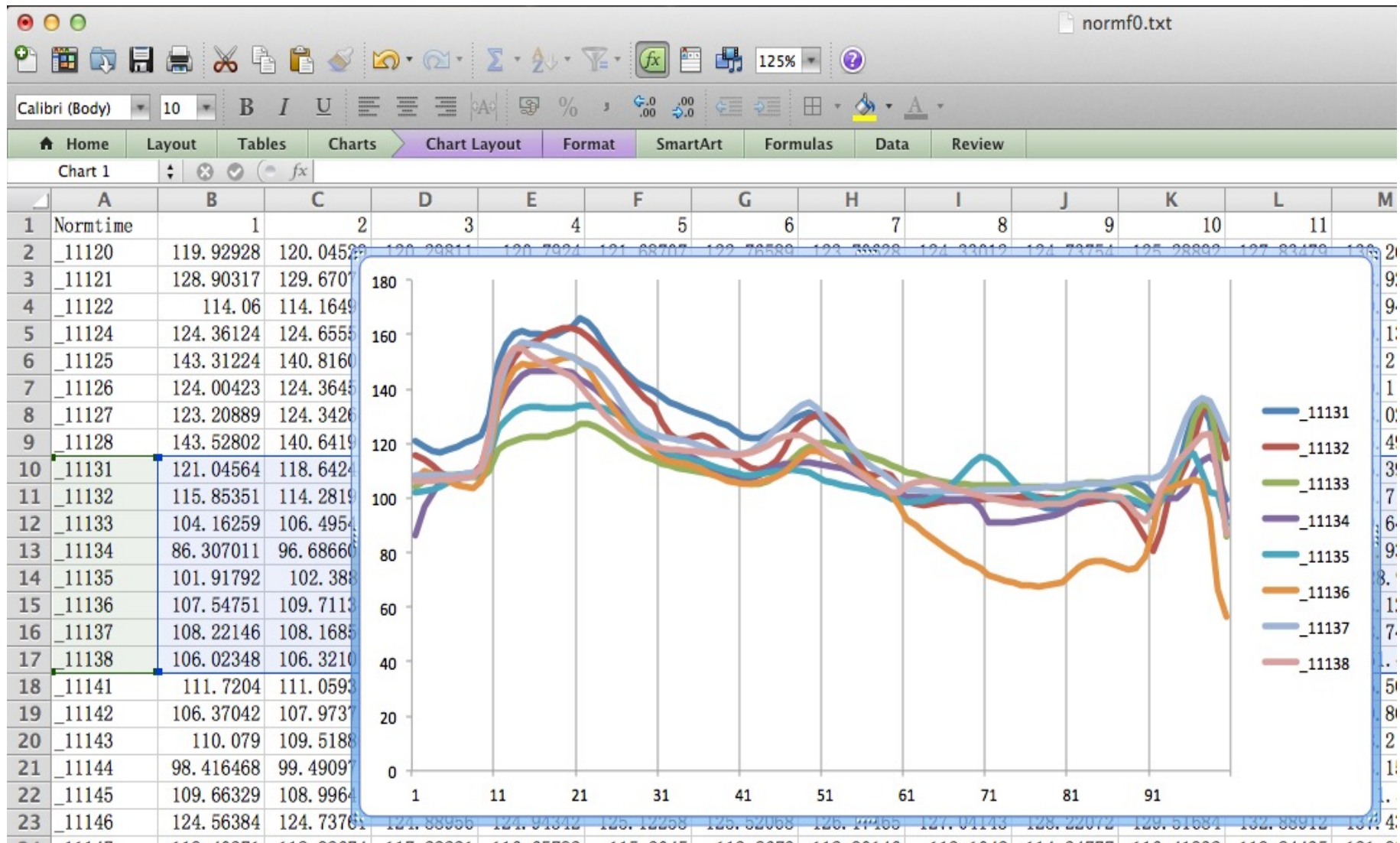
☒ Set initial normalized time to 0

Standards Cancel Apply OK

Save ensemble results by checking "Get ensemble files", and entering the right number of repetitions if applicable

- File Edit Search Convert Font Help
- Ensemble files saved:
- 1) normf0.txt
 - 2) normtime_semitonef0.txt
 - 3) normtime_f0velocity.txt
 - 4) normtimeIntensity.txt
 - 5) normactutime.txt
 - 6) maxf0.txt
 - 7) minf0.txt
 - 8) excursionsize.txt
 - 9) meanf0.txt
 - 10) duration.txt
 - 11) maxvelocity.txt
 - 12) finalvelocity.txt
 - 13) finalf0.txt
 - 14) meanintensity.txt
 - 15) samplef0.txt
 - 16) f0velocity.txt
 - 17) mean_normf0.txt
 - 18) mean_normtime_semitonef0.txt
 - 19) mean_normtime_f0velocity.txt
 - 20) mean_normtimeIntensity.txt
 - 21) mean_normactutime.txt
 - 22) mean_maxf0.txt
 - 23) mean_minf0.txt
 - 24) mean_excursionsize.txt
 - 25) mean_meanf0.txt
 - 26) mean_duration.txt
 - 27) mean_maxvelocity.txt
 - 28) mean_finalvelocity.txt
 - 29) mean_finalf0.txt
 - 30) mean_meanintensity.txt

Plotting “normf0.txt” in Excel



Processing measurement ensemble files in Excel

1	Filename	a	c	e	g	j
2	_m111a	105.65	156.1126	182.6746	212.5252	141.8502
3	_m111b	134	180.9628	202.2626	211.5624	102.905
4	_m111c	106.7626	140.8248	225.5128	215.8124	136.1126
5	_m111d	115.0624	151.9128	181.9374	217.7	139.3772
6	_m113a	124.5008	154.1624	171.2628	185.7208	168.5582
7	_m113b	126.8698	166.2756	176.9596	183.781	163.9298
8	_m113c	112.1454	177.6876	221.8124	192.082	153.3094
9	_m113d	119.888	158.8122	153.9874	190.7546	179.1414
10	_m121a	109.93	164.1626	170.4878	202.75	130.4252
11	_m121b	140.8036	169.7504	168.8624	211.025	121.659
12	_m121c	94.5726	170.3002	243.1626	219.4	132.5874
13	_m121d	109.0748	155.3504	162.4372	213.375	141.1406
14	_m123a	104.1364	152.4876	170.0126	183.3984	184.4202
15	_m123b	135.7622	169.1126	165.3376	174.0002	159.1434
16	_m123c	101.78	167.8752	245.475	194.2986	150.039
17	_m123d	115.312	152.55	158.6626	193.6874	180.6072
18	_m141a	102.6612	159.2878	178.9998	205.55	143.5002
19	_m141b	130.1768	174.5	181.1624	208.8376	133.7262
20	_m141c	103.5546	182.0998	224.1252	230.5626	129.3668
21	_m141d	119.6226	145.5624	175.1878	213.5748	143.5298
22	_m143a	115.6664	161.6498	163.9122	175.3002	184.0762
23	_m143b	128.5912	166.2254	164.6498	174.375	155.645
24	_m143c	111.33	170.2876	224.2752	187.898	158.4668
25	_m143d	110.5508	154.0248	168.025	186.787	185.9942
26	_m211a	134.035	169.7252	182.8496	225.3752	159.3152
27	_m211b	144.9762	197.0624	189.2	208.1628	145.488
28	_m211c	118.7796	192.8252	227.6622	217.6876	127.4604
29	_m211d	135.2938	181.1372	181.3126	218.9	157.91
30	_m213a	135.1422	167.891	180.5964	188.8134	165.303
31	_m213b	148.5732	203.0498	173.3126	187.2468	166.0122
32	_m213c	131.3028	202.1874	225.5126	191.4282	158.638
33	_m213d	125.9904	173.882	167.318	200.1394	171.0962
34	_m221a	120.8734	188.7748	175.675	204.25	148.7126
35	_m221b	153.757	185.5878	175.9624	214.3622	134.6482

E	F
556587	296.79422
581008	220.43842
820279	223.200477
236612	308.408171
958895	275.540174
652131	230.69803
917234	218.455543
360302	283.459535
230114	289.729072
264912	220.760121
515996	234.833919
738391	307.714163
718603	277.681056
162682	231.600934
862751	208.459716
631577	278.736184
5.92825	286.391358
1.77442	227.503634
752967	215.972347
276077	309.212785
339078	280.819096
470577	216.428225
277663	218.968527
795007	287.979752
971479	277.421554
370788	221.842855
411163	213.159619
832931	288.622441
330725	264.208469
928969	216.000091
565429	217.579099
5.51016	265.079892
143939	283.534671
195315	211.064208

E	F
02.222456	300.509675
220.45953	221.807151
79.619393	230.809932
08.493136	310.46379
51.105287	269.91017
93.876552	221.421376
57.943289	220.208978
58.797527	275.154404
73.892119	291.716827
212.06728	218.417204
69.933906	232.219469
85.068041	310.530459
40.236039	267.439302
86.709295	221.066455
62.075626	216.181633
52.324287	266.74631
59.874283	289.974502
15.323513	222.688886
213.37255	214.866681
73.674555	313.046661
10.288753	270.296593
89.308887	215.44689
206.39759	215.821069
11.991333	276.521234
83.030484	278.0311
17.680196	216.209387
54.687668	215.857288
83.711531	289.028849
238.63155	258.433299
00.683136	213.643644
45.400164	216.275719
42.556144	262.604434
62.198142	283.071552
13.776108	213.779875

E	F
05.311.968757	306.560861
09.229.04624	233.214568
04.333.85926	250.694448
09.316.680312	318.495807
07.317.882703	287.094372
05.234.261979	242.518349
02.340.409837	233.387822
04.326.187048	296.236378
07.288.192621	296.76527
01.215.981837	226.300893
04.296.690657	248.144437
07.304.659502	313.691961
04.288.537903	283.304373
01.219.705465	239.047704
03.321.397481	231.394399
04.304.115241	285.631228
06.279.077626	297.175809
09.220.737953	239.289068
09.237.295092	228.471165
01.299.977353	317.833813
07.269.264124	299.288256
02.229.649173	227.958313
09.258.726821	230.986294
09.255.586091	302.982087
01.289.570005	283.035525
01.228.046397	228.926386
02.319.382971	225.675321
02.290.946064	295.298273
02.295.297034	277.337816
05.236.989778	235.568917
07.323.94218	233.51738
06.304.465321	276.834255
09.279.035722	287.474068
05.220.640425	224.728911

mean_duration.txt

mean_finalf0.txt

mean_meanf0.txt

mean_maxf0.txt

Start

Input File No:

Nrepetitions:

TextGrid extension:

Sound file extension:

Or .WAV, .aiff, .AIFF, .mp3, .MP3

Task: ☐ Interactive labeling
☐ Process all sounds without pause
☐ Get emsemble files
☒ Average across speakers

Speaker folder location:

Speaker folder file:

F0 analysis options:

F0 range (Hz):

N. normalized times per interval:

F0 sample rate (Hz):

Perturbation length (s):

Final offset (s):

Smoothing window width (s):

☒ Save output files

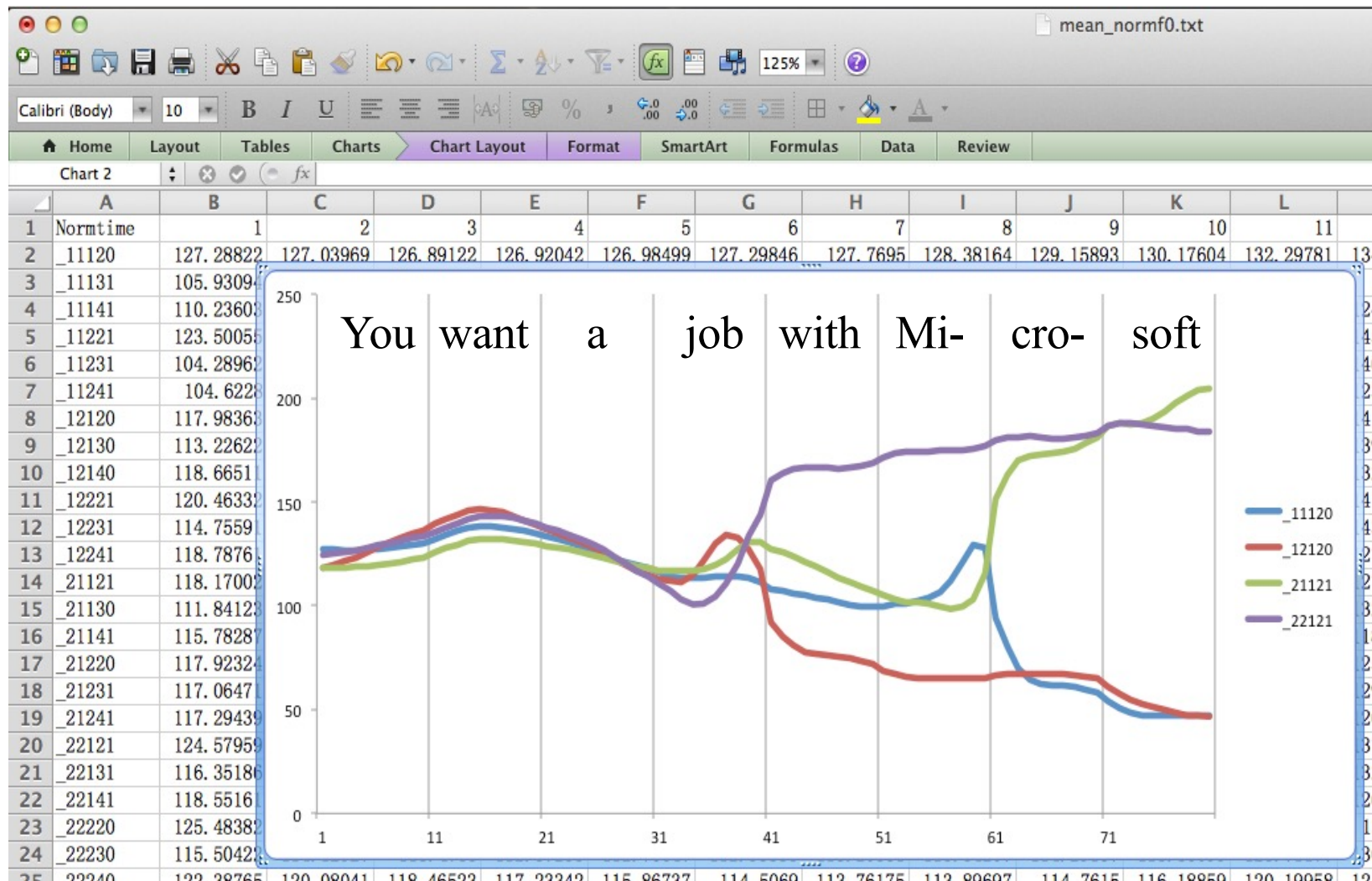
☒ Set initial normalized time to 0

Save cross-speaker normf0 files by checking “Average across speakers”

Average normf0 contours across speakers

1. Create a text file “**speaker_folders.txt**”, which contains the speaker folder names arranged in a single column.
2. Run ProsodyPro with **task 4 — Average across speakers — checked**.
3. The script will read **mean_normf0.txt** from all the speaker folders, average the f0 values on a logarithmic scale, and then convert them back to Hz.
4. In the Start window, you also need to tell ProsodyPro where the speaker folder file is. The default location is the current directory: **“./”**. If it is in an upper directory, you should enter **“../”**
5. The grand averages are saved in **“mean_normf0_cross_speaker.txt”**

Plotting “mean_normf0.txt” in Excel



- Data from Liu et al. (2013), showing interactions of word stress, focus and sentence type

Conclusions

Highlights of ProsodyPro:

1. **A comprehensive analysis tool**
2. **Enables both detailed analysis of continuous prosody, and systematic comparison of discrete measurements**
3. **Minimizes labor by automating tasks that do not require human judgment**
4. **Facilitates human intervention of processes that are prone to error**

Other tools

Available:

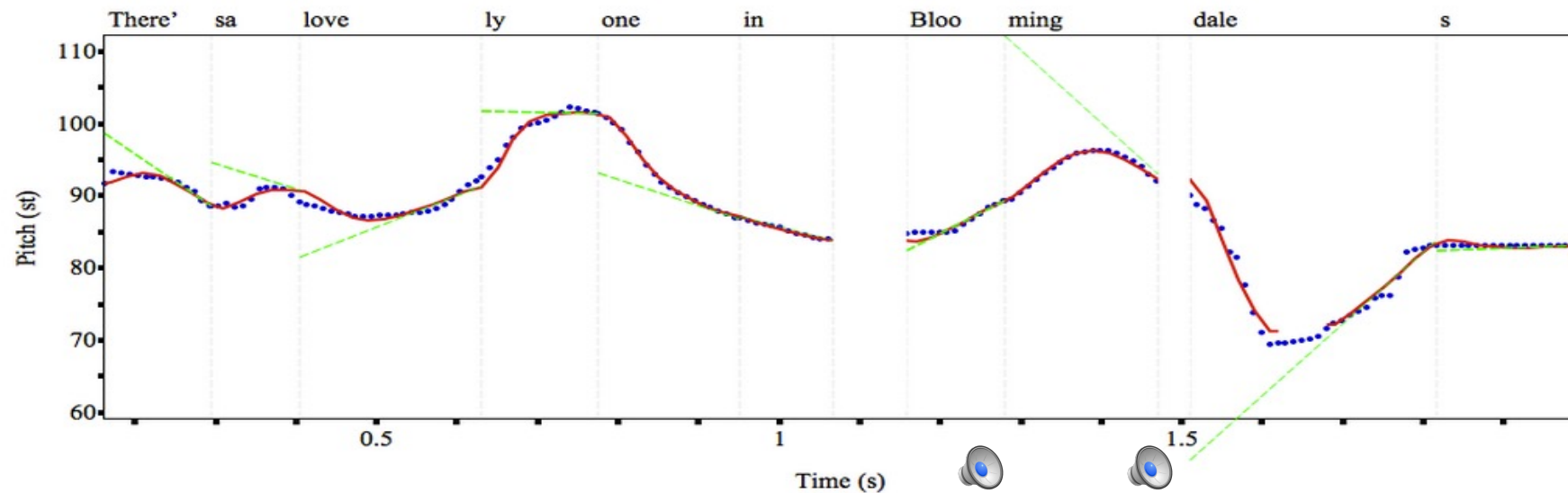
1. **PENTAtainer1** — Analyzing intonation by modeling:
Extracting qTA parameters from individual sentences, via automatic analysis-by-synthesis
2. **PENTAtainer2** — Analyzing intonation by modeling:
Extracting function-specific qTA parameters from an entire speech corpus (see afternoon poster)

To be released soon:

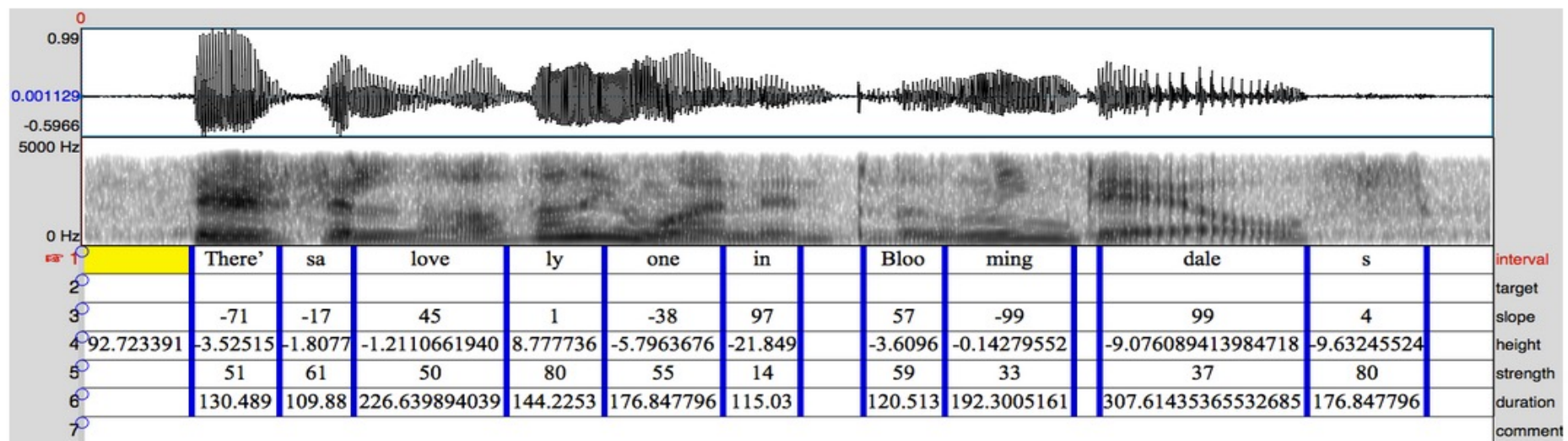
1. **FormantPro** — Analyzing continuous formants just like F_0
2. **ProsodyPro_BID** — Add emotion-related measurements to ProsodyPro

Merci!

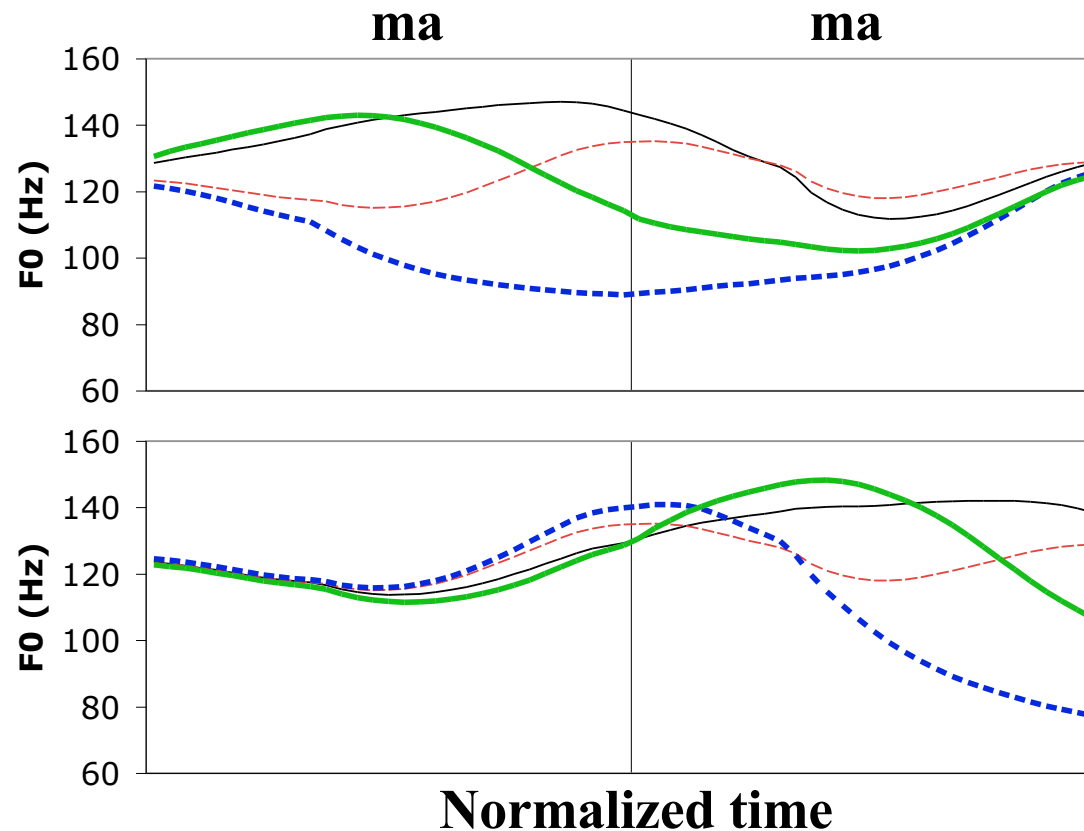
PENTAtainer1—A Praat script for extracting pitch targets from individual sound files



Replot Zoom in Zoom out Zoom all |<< << >> >>| Play original Play resynthesis Previous Next Exit



Experimental control via minimal contrast comparison



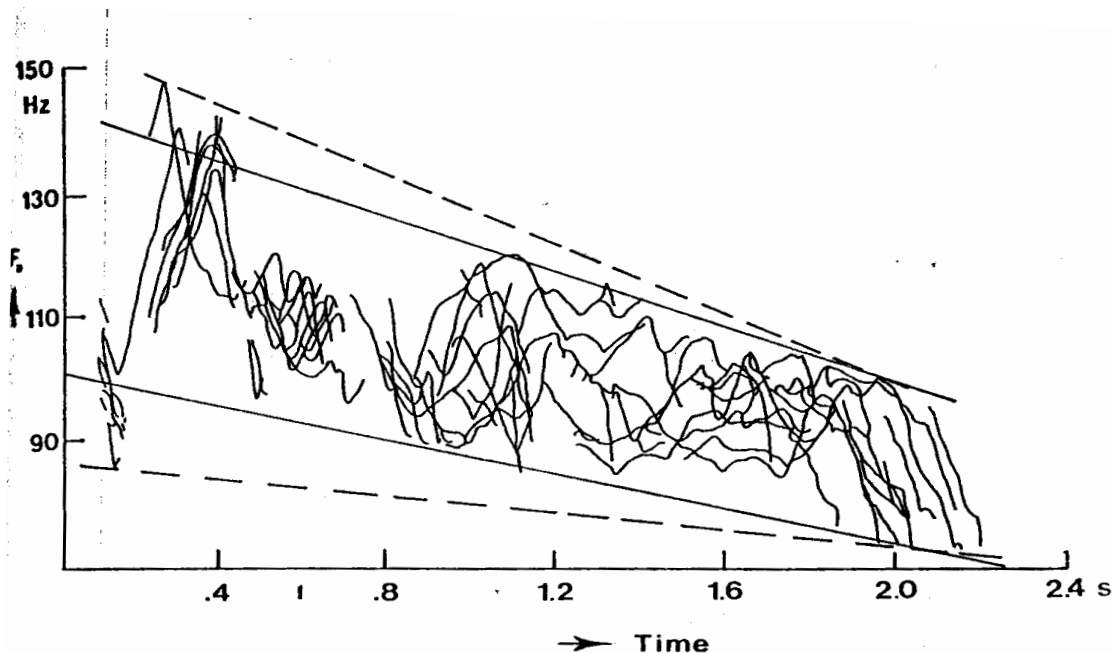
Xu (1997):
*Average of 40
repetitions by 8
subjects*

- ◆ Here all other factors are kept constant while the surrounding tonal contexts are systematically varied
- ◆ And F_0 contours are overlaid on each other in the same plot

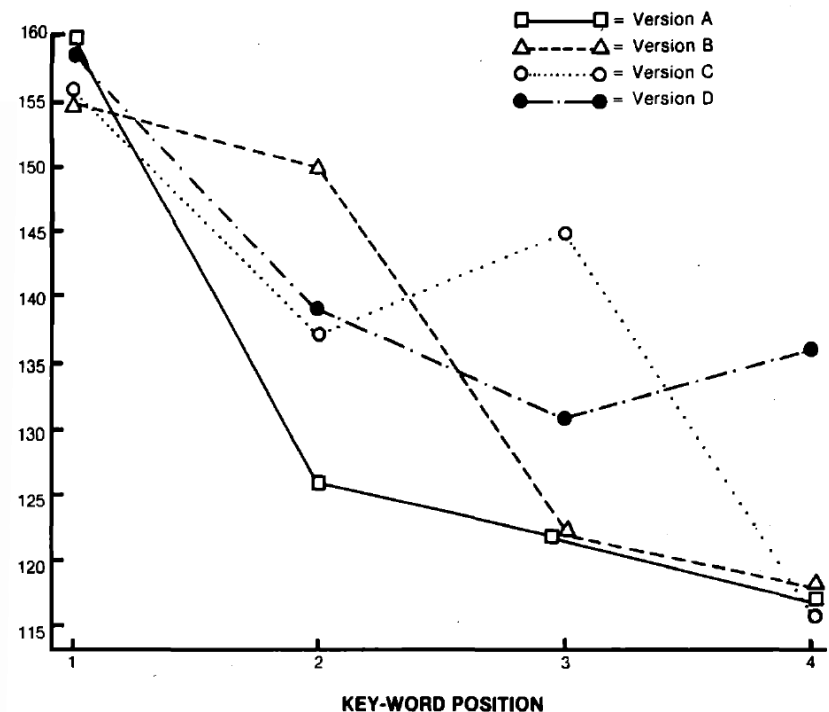
Dilemma in handling a lot of data

- To see all the details, or to pick a few points for systematic statistical comparison?

The *fish* will be *fresh* and *cheap* at this *restaurant*.



(Maeda 1976, cited by Cohen et al. 1982: 257)



Cooper, Eady & Mueller (1985)

- Is it possible to do both?*