Workflow for annotating, aligning, and analyzing files

Raksit T. Lau-Preechathammarach

2019.05.30

This file documents the process for annotating files, aligning them, realigning them, and then analyzing them on VoiceSauce. Each step will be demonstrated with the sample 45 file. You can see any of these files for yourself in the 45sample folder under files.

(A) Annotation

EXCEL FILE INFO

(1) The metadata for each participant may be found in the participant numbered Excel file (ex. 45.xlsx). In **Sheet 1**, please make the following replacements with the Find and Replace option:

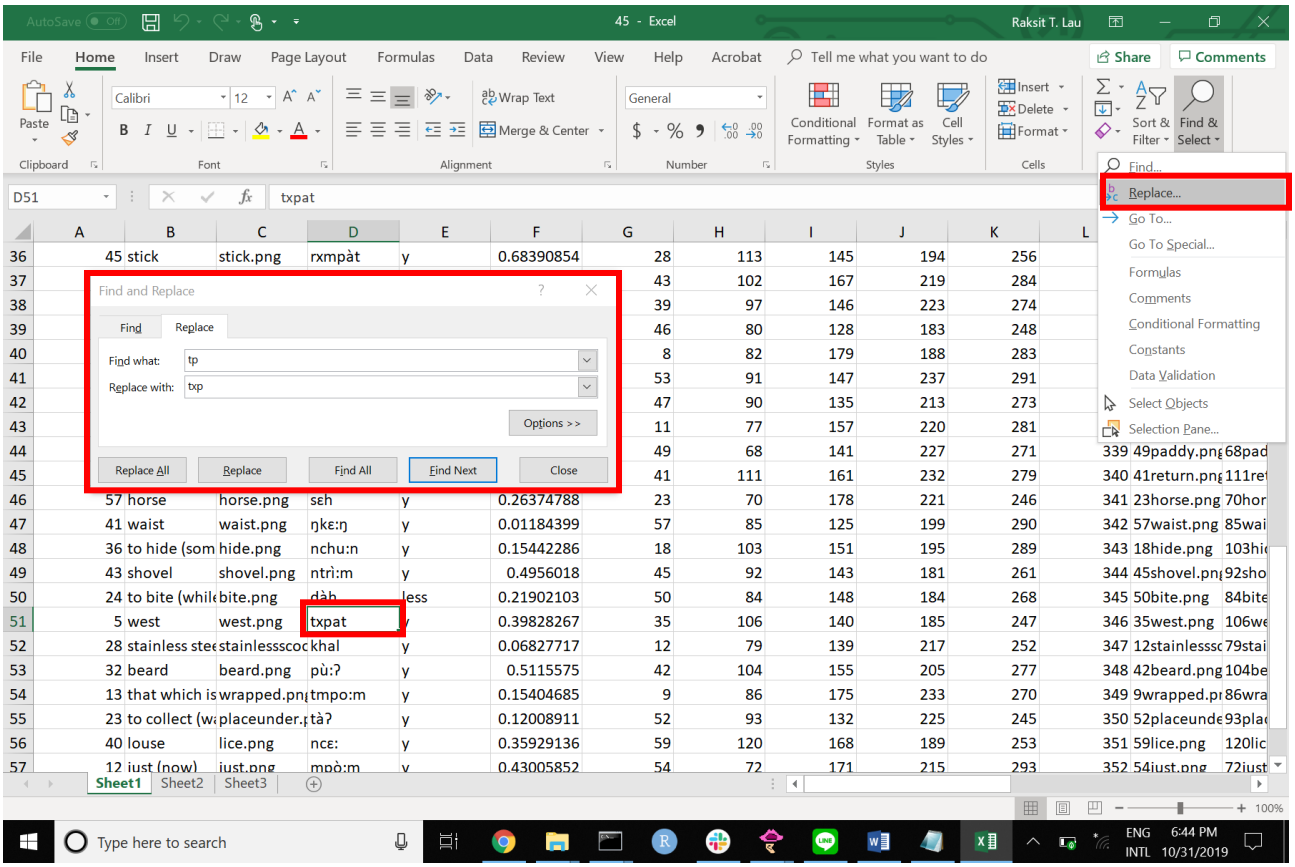
collectwater --> placeunder

ɤ → ə
æ → ɛ

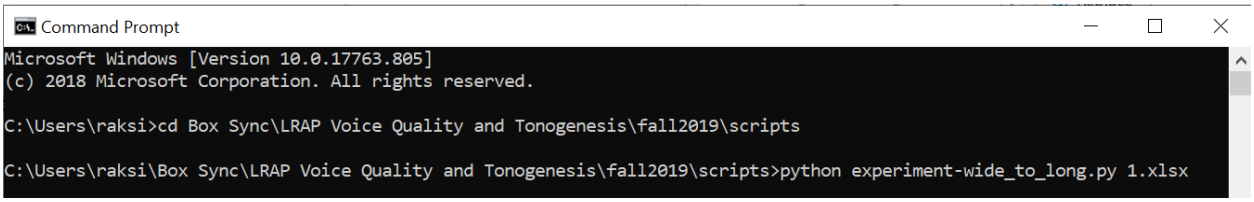
Please account for sesquisyllables by adding x in the following combinations

tp → txp
tm → txm
lm → lxm
rm → rxm
sŋ → sxŋ
kt → kxt
phl → phxl
cn → cxn

In the picture below, you will see that you can find “Replace” in the Home tab (note this may change depending on your version of Excel). When you click it, the Find and Replace window will open up. The box below shows an example of “txpat”, which has already been changed from “tpat”.



(2) This sheet is also in what is called “wide” format. We want to convert it to “long” format. To do this, first close the file. Then, you must use the experiment-wide_to_long.py python script, written by Qingyang Wang. You can run the python script in your command line (search cmd for Windows, use Terminal in Mac). Your first command in the command line should be to cd (change directory) to the folder with the script and then run the python script with the excel file as the first argument. Make sure the file is in the same folder as the script.

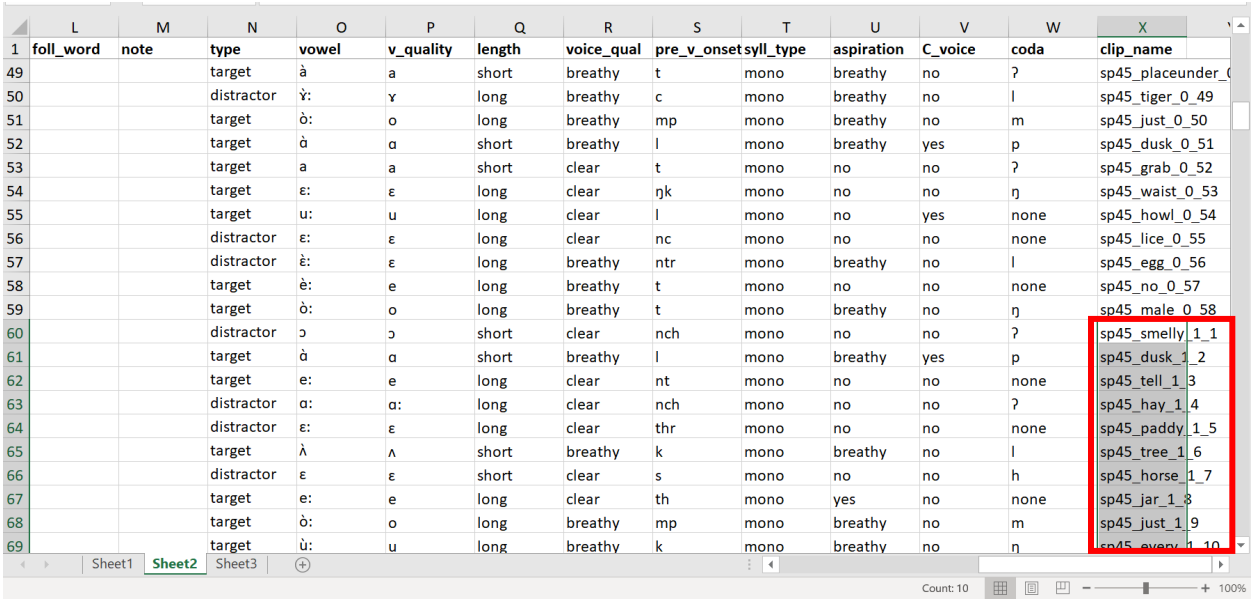


(3) Now Sheet 2 should be populated. For now, ignore Columns I through M (participant responses, before and after, etc.). These will have to be filled out by hand later.

(4) The clip_name (the rightmost column in Sheet 2) is important for labeling the files correctly later. The format of the clip name is:

[speaker number]_[word]_[round]_[order within round]

Copy everything from the clip_name column from the **first row of Round 1 (IGNORE ROUND 0)** down through Round 5 and paste it in a text file within the speaker folder and label it "clips.txt".



	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	foll_word	note	type	vowel	v_quality	length	voice_qual	pre_v_onset	syll_type	aspiration	C_voice	coda	clip_name
49			target	à	a	short	breathy	t	mono	breathy	no	ʔ	sp45_placeunder_(
50			distractor	ɤ̀ː	ɤ	long	breathy	c	mono	breathy	no	l	sp45_tiger_0_49
51			target	òː	o	long	breathy	mp	mono	breathy	no	m	sp45_just_0_50
52			target	ɑ̀	ɑ	short	breathy	l	mono	breathy	yes	p	sp45_dusk_0_51
53			target	a	a	short	clear	t	mono	no	no	ʔ	sp45_grab_0_52
54			target	ɛː	ɛ	long	clear	ŋk	mono	no	no	ŋ	sp45_waist_0_53
55			target	uː	u	long	clear	l	mono	no	yes	none	sp45_howl_0_54
56			distractor	ɛː	ɛ	long	clear	nc	mono	no	no	none	sp45_lice_0_55
57			distractor	ɛ̀ː	ɛ	long	breathy	ntr	mono	breathy	no	l	sp45_egg_0_56
58			target	èː	e	long	breathy	t	mono	no	no	none	sp45_no_0_57
59			target	òː	o	long	breathy	t	mono	breathy	no	ŋ	sp45_male_0_58
60			distractor	ɔ	ɔ	short	clear	nch	mono	no	no	ʔ	sp45_smelly_1_1
61			target	ɑ̀	ɑ	short	breathy	l	mono	breathy	yes	p	sp45_dusk_1_2
62			target	eː	e	long	clear	nt	mono	no	no	none	sp45_tell_1_3
63			distractor	ɑː	ɑː	long	clear	nch	mono	no	no	ʔ	sp45_hay_1_4
64			distractor	ɛː	ɛ	long	clear	thr	mono	no	no	none	sp45_paddy_1_5
65			target	ʌ̀	ʌ	short	breathy	k	mono	breathy	no	l	sp45_tree_1_6
66			distractor	ɛ	ɛ	short	clear	s	mono	no	no	h	sp45_horse_1_7
67			target	eː	e	long	clear	th	mono	yes	no	none	sp45_jar_1_8
68			target	òː	o	long	breathy	mp	mono	breathy	no	m	sp45_just_1_9
69			target	ùː	u	long	breathy	k	mono	breathy	no	ŋ	sp45_every_1_10

Sheet1 Sheet2 Sheet3

Count: 10 100%

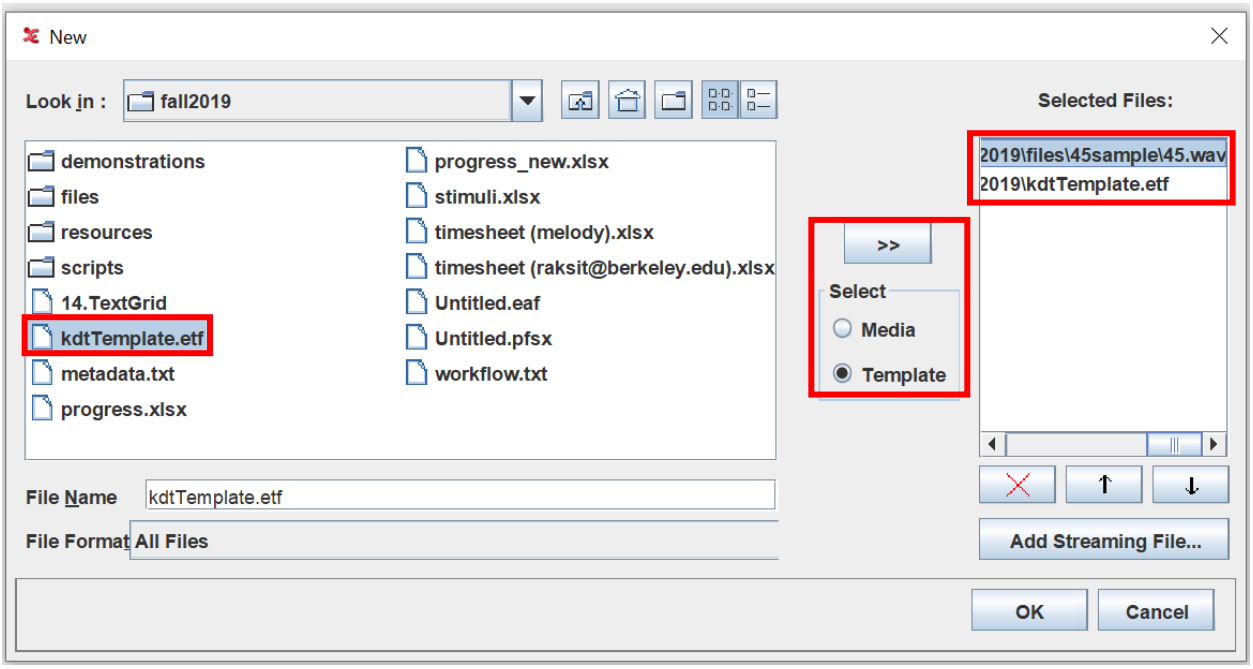
clips.txt - Notepad

File Edit Format View Help

```
sp45_smelly_1_1
sp45_dusk_1_2
sp45_tell_1_3
sp45_hay_1_4
sp45_paddy_1_5
sp45_tree_1_6
sp45_horse_1_7
sp45_jar_1_8
sp45_just_1_9
sp45_every_1_10
sp45_divorce_1_11
sp45_sensitive_1_12
sp45_lightning_1_13
sp45_side_1_14
sp45_coconut_1_15
sp45_stainlessscooper_1_16
sp45_toomuch_1_17
sp45_eggplant_1_18
sp45_swelling_1_19
sp45_male_1_20
sp45_bite_1_21
sp45_waist_1_22
sp45_wrapped_1_23
sp45_cow_1_24
sp45_grab_1_25
sp45_egg_1_26
sp45_toastedrice_1_27
sp45_tiger_1_28
sp45_shovel_1_29
sp45_placeunder_1_30
sp45_spin_1_31
sp45_slap_1_32
sp45_shoulder_1_33
sp45_five_1_34
sp45_tall_1_35
sp45_sow_1_36
sp45_old_1_37
sp45_six_1_38
sp45_redant_1_39
sp45_hide_1_40
```

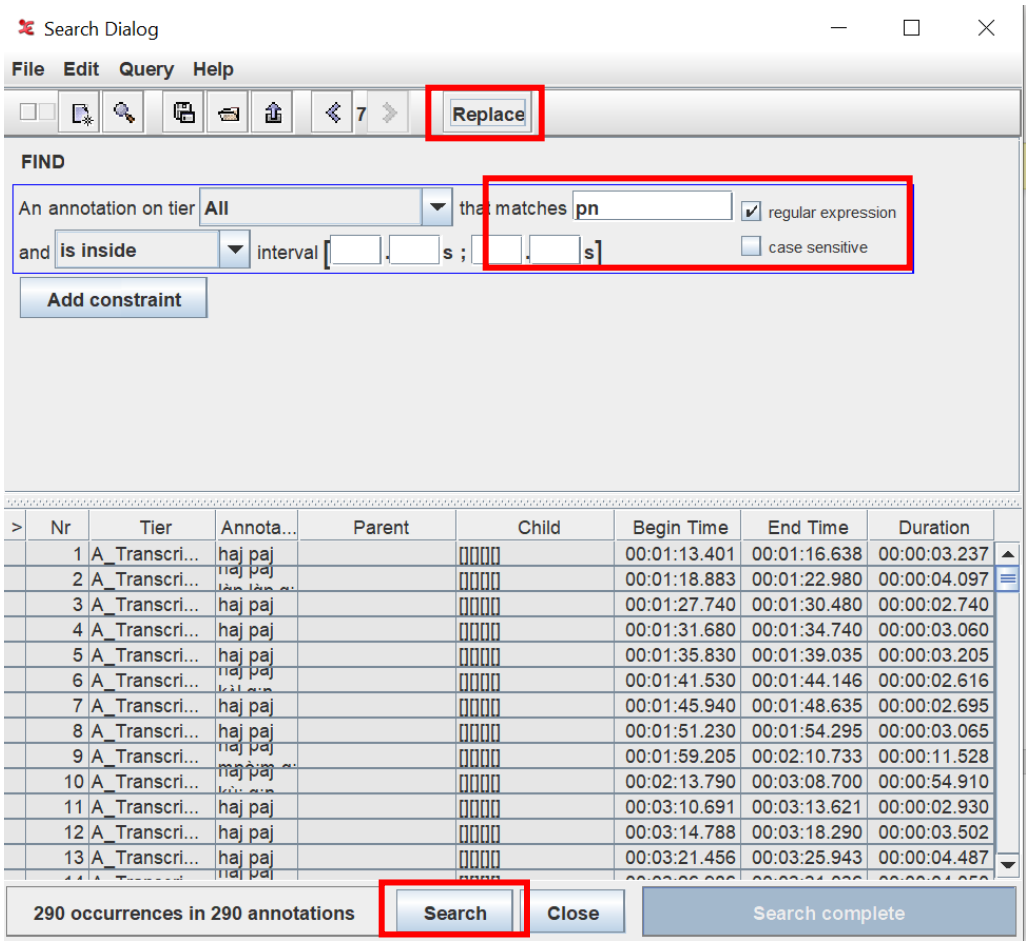
Segmentation in ELAN

(1) Open ELAN and go to File > New. Select the participant .wav file as the **Media** and the kdt-template.etf file as the **template**. Then press OK.



(2) If you are working with an ELAN file that has been partially filled in (most likely by me), you should check and make the following replacements. These are the same changes as in the Excel file plus a few extra ones just in case (not all of these will be found in every file). To make the changes, go to Search > Find (and Replace)… Under "Find", press "An annotation". You can then search the sequence to replace (make sure you check off "regular expression"). When you click "Search", you will get all the results. If there are some, you can then click "Replace" at the top to replace the sequence.

ɤ → ə
æ → ɛ
tp → txp
tm → txm
lm → lxm
rm → rxm
sŋ → sxŋ
kt → kxt
phl → phxl
cn → cxn
pnaj → pxnaj
cŋat → cxŋat
nk → ŋk
lp → lxp
rp → rxp
sl → sxl
st → sxt
sk → sxk
tl → txl
tb → txb
tŋ → txŋ
pd → pxd



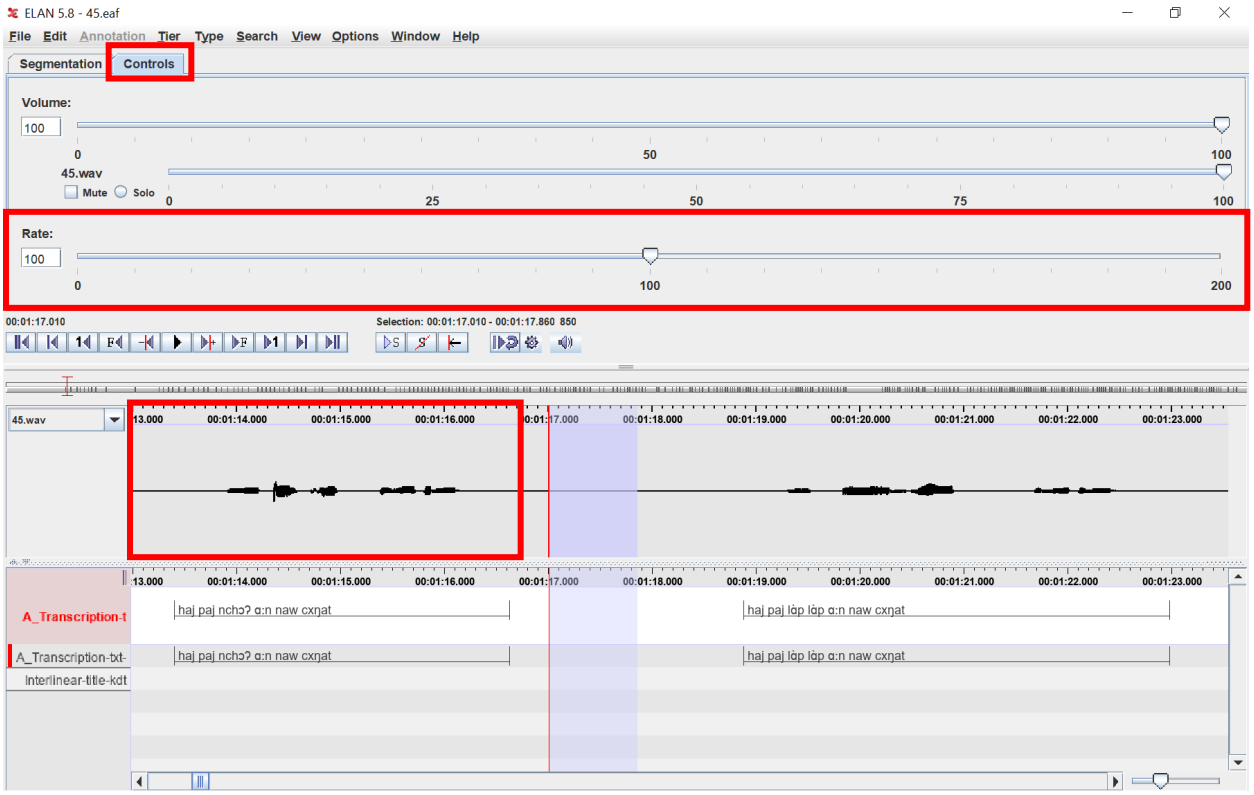
(3) Now you will segment the files. Go to **Options > Segmentation mode**. Check the speaker excel file for what the **first word of round 1** is. Then, search around the beginning of the file for the first sentence. You should hear roughly the same sentence (we will call this the **carrier sentence**) with only one word, the **target word**, replaced each time. The sentence should sound roughly like:

haj wa:w pxnaj [TARGET] ɑ:n naw cxŋat

Other words that may appear:
siŋ 'will (future)'
(n)te: 'to say'
muaŋ 'you'
paj '(quotative marker)'
tɔ: nɛ̀: 'this one'

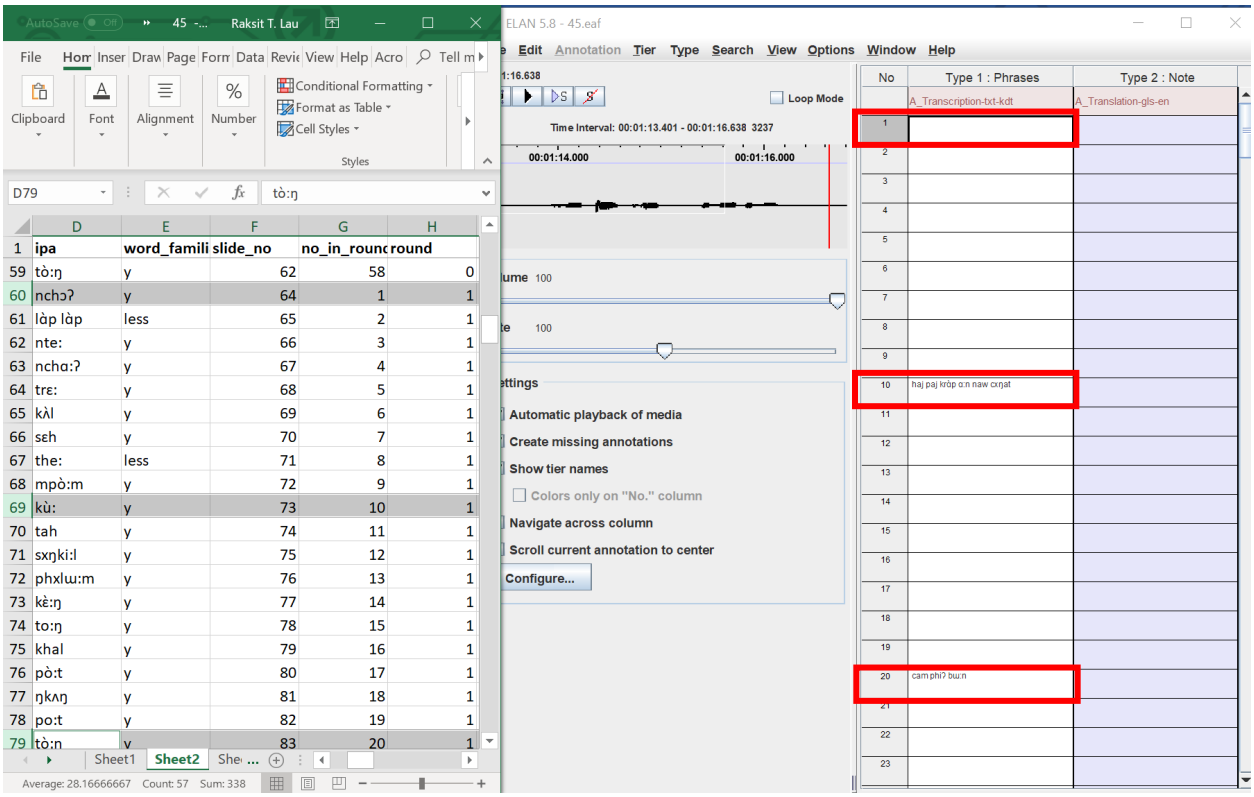
Note that for the speaker in the screenshot below, the speaker says /paj/ instead of /pxnaj/. The waveform for each sentence should also look roughly the same, which is a hint for segmentation. Segment each of the sentences out. To speed up the process, you can go to increase the Rate in the Controls tab to 200.

You can press **Ctrl+Space** to start and stop playing from the cursor. You can press **Enter** to set a boundary and **Enter** again to set the other side of the boundary. If you need to delete an annotation, you can right click and delete it with Remove Annotation.

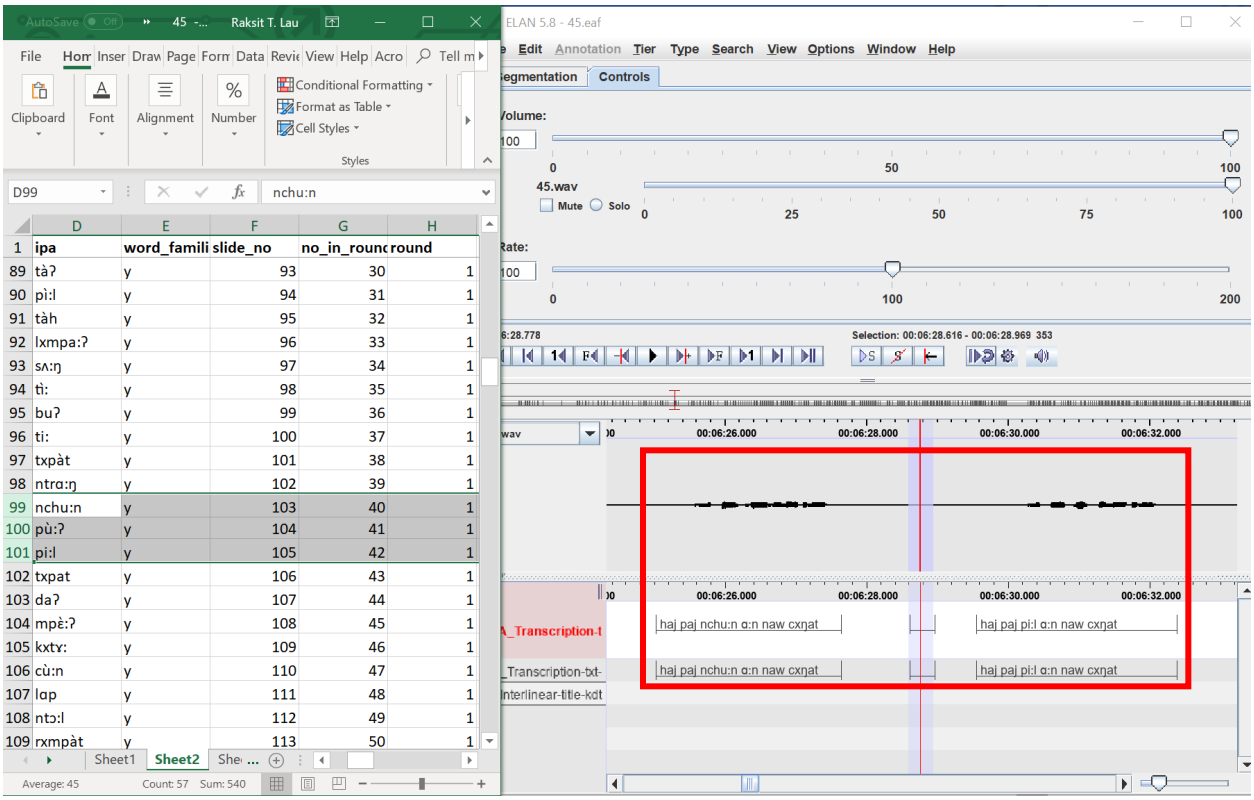


(4) Now go to **Options > Transcription mode.** Select "Phrases" for Column 1 and "Note" for Column 2. Then press **Apply**. We want there to be 290 segments. Chances are there won't be. Listen to every 10 sentences (or so—it is up to you—the main point of this step is to make sure we end up with 290 segments). and cross-reference with every 10 or so sentences in the excel file (starting from round 1, no_in_round 1). I like to set the ELAN and Excel windows side-by-side for ease of viewing. If they say a different word, check if it is a different (roughly neighboring) target word or one of the potential alternatives to the target word (see list below). If they say the correct word, you don't have to fill in anything.

As an example, let's say we listen to segment 1 and hear [nchɔʔ]. Then we can leave it blank. We then listen to segment 10 and hear [kràp]. It is not a nearby word (i.e. not the:, or mpò:m, or tah, for example) but it is in the list for alternative words for [kù:]. So we write it in with the carrier sentence. If you hear a word that is not any other word in any wordlist, you can consult me to ask what the word is or just leave it blank for now (it is not very important at this stage). We then listen to segment 20 and hear [cam phiʔ bɯ:n], which is in the list of sentences speakers may use when they don't remember the word, so we write that in as well.



If a segment doesn't match up with the word and instead matches up with a nearby one, it probably means that there was a word that was skipped or there is an extra one. If a word was skipped, just make a blank segment as a placeholder.



	ipa	word_famili	slide_no	no_in_round	round
89	tàʔ	y	93	30	1
90	pì:l	y	94	31	1
91	tàh	y	95	32	1
92	lxmpa:ʔ	y	96	33	1
93	sʌ:ŋ	y	97	34	1
94	tì:	y	98	35	1
95	buʔ	y	99	36	1
96	ti:	y	100	37	1
97	txpàt	y	101	38	1
98	ntrɑ:ŋ	y	102	39	1
99	nchu:n	y	103	40	1
100	pù:ʔ	y	104	41	1
101	pi:l	y	105	42	1
102	txpat	y	106	43	1
103	daʔ	y	107	44	1
104	mpɛ̀:ʔ	y	108	45	1
105	kxtʏ:	y	109	46	1
106	cù:n	y	110	47	1
107	lɑp	y	111	48	1
108	ntɔ:l	y	112	49	1
109	rxmpàt	y	113	50	1

Notice here that there [pù:ʔ] should be between [nchu:n] and [pi:l], but that the speaker did not say anything in between. Thus, we create an arbitrary segment in the blank space between the two words but do not fill it in with anything. If there is something said in between, however, you can segment that audio.

Common words you may hear in place of specific target words:

1) lu: 'to howl'	hɔ:n (the Thai word), krɑh 'to bark'
2) lù: 'thigh'	kò:n lù: (compound word for 'thigh'), jɯ:ŋ 'foot', cxŋkàŋ 'shin'
3) kù: 'every'	kràp 'each', tùk 'suffering', txŋaj 'day'
4) txpat 'west'	ntiàŋpat 'west', ntiàŋ ntruh 'sunset'
5) txpàt 'six'	hɑk 'to fall', ntrùh 'to fall'
6) làp làp 'dusk'	txbɯ: 'evening'
7) kho:kho 'toasted rice'	kxntiàʔ 'deep fried'
8) pho:m 'fragrant'	hu:ɲ 'to sniff'
9) mpò:m 'just (now)'	sxʔa:p 'to yawn', adɯh 'before', mpò:m ju:l 'just got up'
10) txmpo:m 'wrapped'	to:m 'to wrap'
11) tè: 'no'	phiʔ aj 'is not', tè:tè: 'no'
12) the: 'jar'	sxlu:ɲ 'jar'
13) po:t 'swelling'	plo:h 'bruise'
14) pò:t 'too much'	da:l 'a lot', da:l pò:t 'too much'
15) tàʔ 'to place under'	rɔ̀:ŋ (the Thai word)
16) dàh 'to bite'	kap ɑ:n txli: 'bite until break', txli: 'break', kap 'bite'

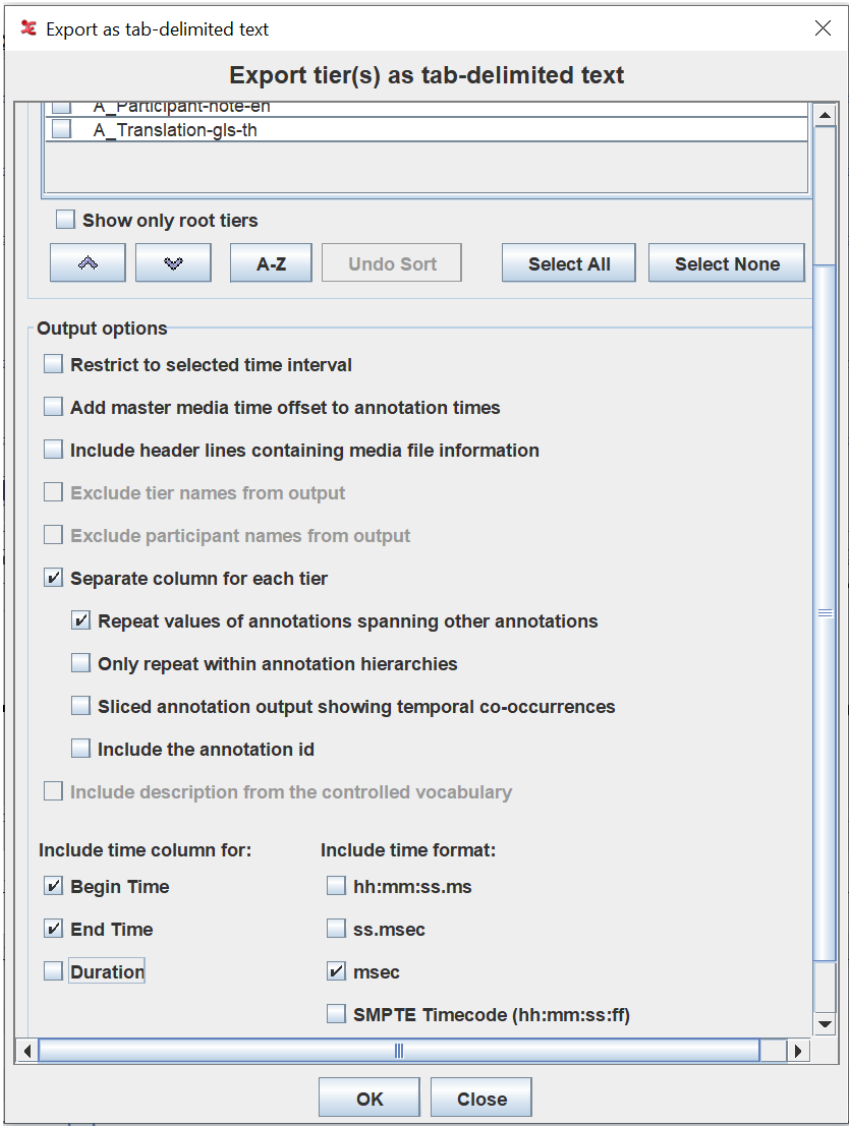
17) tah 'to break up' riàŋ 'divorce' (Thai)
18) khaI 'bowl' kxpɑːŋ 'bucket'
19) toːŋ 'coconut' plaj toːŋ 'coconut fruit'
20) tòːŋ 'male' tɑː tòːŋ 'male', tòːl 'male'
21) puːʔ 'sun ray' kxtaw 'hot', ntiàŋ 'sun', txŋaj 'day'
22) ùːʔ 'beard' ɲah 'curly'
23) buʔ 'to sow' prʌ̀h 'to sow'
24) piːl 'flower' kʌ̀l piːl 'flower'
25) pìːl 'to roll, wind' mùːl 'to spin', pàn 'to spin', muàn 'to spin'
26) nchuːn 'to send' abuŋ 'spoon'
27) mpèːʔ 'mom' meʔ 'mom'
28) ŋkɛːŋ 'waist'
29) ŋkʌŋ 'eggplant' plaj ŋkʌŋ 'eggplant'
30) ntrɑːŋ 'red ant' sxmoːc (kxsaw) '(red) ant'
31) rxmpàt 'stick' aluàŋ 'branch'
32) ntɔːl 'star' lxmʌ̀l 'cloud'
33) kxtɤ 'season' rɯdu: ‘season’ (Thai)
34) sʌːŋ 'five' buaj 'search'
35) phxlɯːm 'lightning' ntàh 'lightning', phatlɯːm 'lightning'
36) sxŋkiːl 'sensitive'
37) kɛ̀ːŋ 'side' kɛ̀ːm 'side'

If the speaker can’t remember the word, they will usually say a few possible things:

- cam phiʔ bɯːn (Kuy for "I can't remember")
- phiʔ / phuʔ diŋ (Kuy for "I don't know")
- phàːn (Thai for "pass")

(5) Now, export with File > Export as > Tab-delimited Text

Check off only the transcription tier (**A_Transcription-txt-kdt**). Scroll down and check off "Separate column for each tier", "Repeat values of annotations spanning other annotations", "begin time", "end time", and "msec" and then press OK to export. The encoding should be UTF-8.



Create a new excel file (temporary, we can get rid of this later) and copy and paste the tab-delimited text file into it. You should see three columns labeled **Begin Time, End Time,** and **A_Transcription-txt-kdt**. The last column will be mostly blank but will have any transcriptions you made.

	A	B	C	D	E
1	Begin Tim	End Time	A_Transcription-txt-kdt		
2	73401	76638			
3	78883	82980			
4	87740	90480			
5	91680	94740			
6	95830	99035			
7	101530	104146			
8	105940	108635			
9	111230	114295			
10	119205	130733			
11	133790	188700	haj paj kràp ɑ:n naw cxŋat		
12	190691	193621			
13	194788	198290			
14	201456	205943			
15	206986	211036			
16	214421	218291			
17	222076	226628			
18	231566	236336			
19	239316	242318			
20	245721	248471			
21	251466	254216	cam phiʔ bɯ:n		

Create three new columns, labeled **Before, IPA,** and **After**. In the first row, write everything in the carrier sentence that precedes the target word under the **Before** column (leave a space at the end) and everything that follows under the **After** column (leave a space at the beginning). Go back to the speaker excel sheet and copy from **Sheet 2** everything in the **ipa** column starting from Round 1 (left image). Paste it into the **IPA** column in the temporary excel sheet (right image).

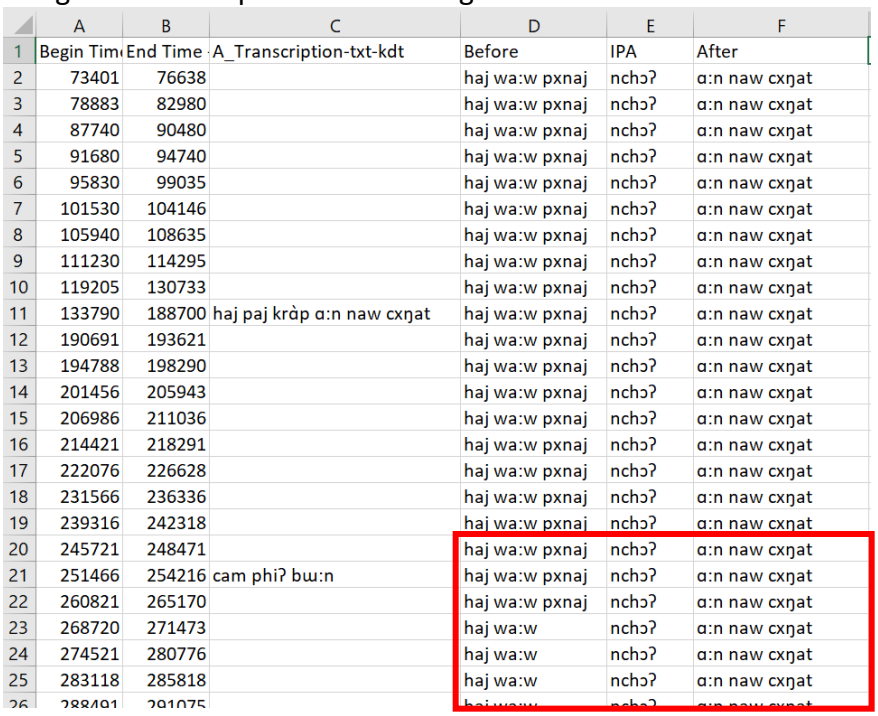
	D	E	F	G	H
1	**ipa**	**word_famili**	**slide_no**	**no_in_rounc**	**round**
56	ncɛ:	y	59	55	0
57	ntrè:l	y	60	56	0
58	tè:	y	61	57	0
59	tò:ŋ	y	62	58	0
60	nchɔʔ	y	64	1	1
61	làp làp	less	65	2	1
62	nte:	y	66	3	1
63	nchɑ:ʔ	y	67	4	1
64	trɛ:	y	68	5	1
65	kàl	y	69	6	1
66	sɛh	y	70	7	1
67	the:	less	71	8	1
68	mpò:m	y	72	9	1
69	kù:	y	73	10	1
70	tah	y	74	11	1
71	sxŋki:l	y	75	12	1
72	phxlɯ:m	y	76	13	1
73	kè:ŋ	y	77	14	1
74	to:ŋ	y	78	15	1
75	khal	y	79	16	1
76	pò:t	y	80	17	1

Sheet1 | Sheet2 | Sheet3

	A	B	C	D	E	F
1	Begin Tim	End Time	A_Transcription-txt-kdt	Before	IPA	After
2	73401	76638		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
3	78883	82980			làp làp	
4	87740	90480			nte:	
5	91680	94740			nchɑ:ʔ	
6	95830	99035			trɛ:	
7	101530	104146			kàl	
8	105940	108635			sɛh	
9	111230	114295			the:	
10	119205	130733			mpò:m	
11	133790	188700	haj paj kràp ɑ:n naw cxŋat		kù:	
12	190691	193621			tah	
13	194788	198290			sxŋki:l	
14	201456	205943			phxlɯ:m	
15	206986	211036			kè:ŋ	
16	214421	218291			to:ŋ	
17	222076	226628			khal	
18	231566	236336			pò:t	
19	239316	242318			ŋkʌŋ	
20	245721	248471			po:t	
21	251466	254216	cam phiʔ bɯ:n		tò:ŋ	
22	260821	265170			dàh	
23	268720	271473			ŋkɛ:ŋ	
24	274521	280776			tmpo:m	
25	283118	285818			kò:	

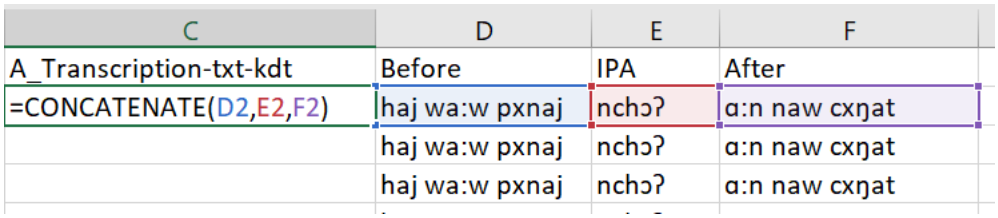
Then highlight the columns “Before, IPA, After” in row 1 and drag the bottom right corner **down** to fill in everything automatically. The result should look like the image below. If the speaker changes the carrier sentence regularly at some point (this happens not uncommonly), then change the “before” or “after” part as necessary and continue to drag the three columns down.

If just a few carrier sentences are off, don't worry about fixing them. Notice in the below image that the speaker changes the beginning of the carrier sentence from /haj wa:w pxnaj/ to just /haj wa:w/, so we change the before part from that segment on and then continue to drag down.



	A	B	C	D	E	F
1	Begin Tim	End Time	A_Transcription-txt-kdt	Before	IPA	After
2	73401	76638		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
3	78883	82980		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
4	87740	90480		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
5	91680	94740		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
6	95830	99035		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
7	101530	104146		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
8	105940	108635		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
9	111230	114295		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
10	119205	130733		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
11	133790	188700	haj paj krɔ̀p ɑ:n naw cxŋat	haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
12	190691	193621		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
13	194788	198290		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
14	201456	205943		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
15	206986	211036		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
16	214421	218291		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
17	222076	226628		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
18	231566	236336		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
19	239316	242318		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
20	245721	248471		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
21	251466	254216	cam phiʔ bɯ:n	haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
22	260821	265170		haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
23	268720	271473		haj wa:w	nchɔʔ	ɑ:n naw cxŋat
24	274521	280776		haj wa:w	nchɔʔ	ɑ:n naw cxŋat
25	283118	285818		haj wa:w	nchɔʔ	ɑ:n naw cxŋat
26	288491	291075				

(6) Now, we want to automatically fill in the A_Transcription-txt-kdt column. In the space in the first row, type the following formula: =CONCATENATE(D2,E2,F2) in the A_transcription-txt-kdt column. You can select the cells instead of typing them in as well. Then, like above, drag the bottom right corner of cell C2 down to fill in the rest automatically. Skip the transcriptions that you already filled in.

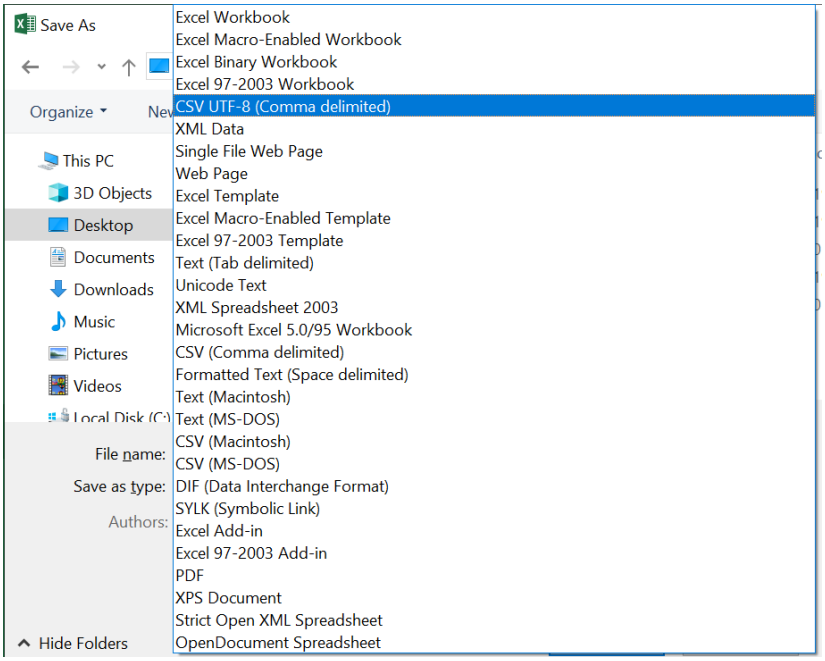


C	D	E	F
A_Transcription-txt-kdt	Before	IPA	After
=CONCATENATE(D2,E2,F2)	haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
	haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
	haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat

Notice that row 11 was previously filled in by us. Thus, we only drag down to the row before it. You can then copy cell C10 and paste in cell C12 (it will copy the *formula*) and continue the dragging process. We stop and repeat this again for cell C21.

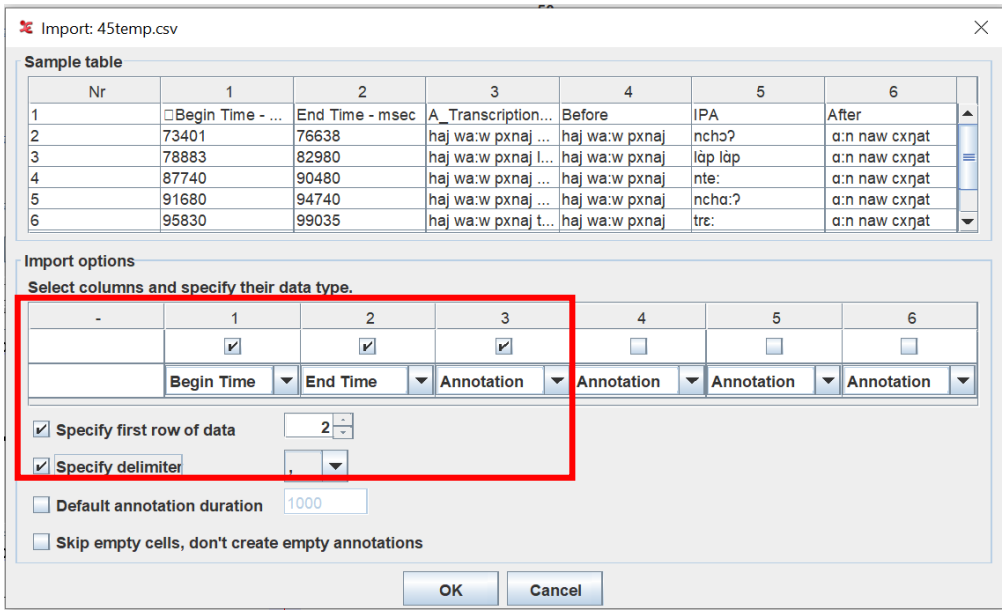
	A	B	C	D	E	F
1	Begin Tim	End Time	A_Transcription-txt-kdt	Before	IPA	After
2	73401	76638	haj wa:w pxnaj nchɔʔ ɑ:n naw cxŋat	haj wa:w pxnaj	nchɔʔ	ɑ:n naw cxŋat
3	78883	82980	haj wa:w pxnaj làp làp ɑ:n naw cxŋat	haj wa:w pxnaj	làp làp	ɑ:n naw cxŋat
4	87740	90480	haj wa:w pxnaj nte: ɑ:n naw cxŋat	haj wa:w pxnaj	nte:	ɑ:n naw cxŋat
5	91680	94740	haj wa:w pxnaj nchɑ:ʔ ɑ:n naw cxŋat	haj wa:w pxnaj	nchɑ:ʔ	ɑ:n naw cxŋat
6	95830	99035	haj wa:w pxnaj trɛ: ɑ:n naw cxŋat	haj wa:w pxnaj	trɛ:	ɑ:n naw cxŋat
7	101530	104146	haj wa:w pxnaj kʌ̀l ɑ:n naw cxŋat	haj wa:w pxnaj	kʌ̀l	ɑ:n naw cxŋat
8	105940	108635	haj wa:w pxnaj sɛh ɑ:n naw cxŋat	haj wa:w pxnaj	sɛh	ɑ:n naw cxŋat
9	111230	114295	haj wa:w pxnaj the: ɑ:n naw cxŋat	haj wa:w pxnaj	the:	ɑ:n naw cxŋat
10	119205	130733	haj wa:w pxnaj mpò:m ɑ:n naw cxŋat	haj wa:w pxnaj	mpò:m	ɑ:n naw cxŋat
11	133790	188700	haj paj krà̀p ɑ:n naw cxŋat	haj wa:w pxnaj	kù:	ɑ:n naw cxŋat
12	190691	193621	haj wa:w pxnaj tah ɑ:n naw cxŋat	haj wa:w pxnaj	tah	ɑ:n naw cxŋat
13	194788	198290	haj wa:w pxnaj sxŋki:l ɑ:n naw cxŋat	haj wa:w pxnaj	sxŋki:l	ɑ:n naw cxŋat
14	201456	205943	haj wa:w pxnaj phxlɯ:m ɑ:n naw cxŋat	haj wa:w pxnaj	phxlɯ:m	ɑ:n naw cxŋat
15	206986	211036	haj wa:w pxnaj kɛ̀:ŋ ɑ:n naw cxŋat	haj wa:w pxnaj	kɛ̀:ŋ	ɑ:n naw cxŋat
16	214421	218291	haj wa:w pxnaj to:ŋ ɑ:n naw cxŋat	haj wa:w pxnaj	to:ŋ	ɑ:n naw cxŋat
17	222076	226628	haj wa:w pxnaj khal ɑ:n naw cxŋat	haj wa:w pxnaj	khal	ɑ:n naw cxŋat
18	231566	236336	haj wa:w pxnaj pò:t ɑ:n naw cxŋat	haj wa:w pxnaj	pò:t	ɑ:n naw cxŋat
19	239316	242318	haj wa:w pxnaj ŋkʌŋ ɑ:n naw cxŋat	haj wa:w pxnaj	ŋkʌŋ	ɑ:n naw cxŋat
20	245721	248471	haj wa:w pxnaj po:t ɑ:n naw cxŋat	haj wa:w pxnaj	po:t	ɑ:n naw cxŋat
21	251466	254216	cam phiʔ bɯ:n	haj wa:w pxnaj	tò:ŋ	ɑ:n naw cxŋat
22	260821	265170	haj wa:w pxnaj dàh ɑ:n naw cxŋat	haj wa:w pxnaj	dàh	ɑ:n naw cxŋat
23	268720	271473	haj wa:w ŋkɛ:ŋ ɑ:n naw cxŋat	haj wa:w	ŋkɛ:ŋ	ɑ:n naw cxŋat

(7) Once you are done filling in all the A_Transcription-txt-kdt cells, you may want to save the Excel file just to make sure nothing went wrong (if you are confident nothing went wrong, you do not need to save it). Now, export this file as a CSV delimited text file UTF-8.

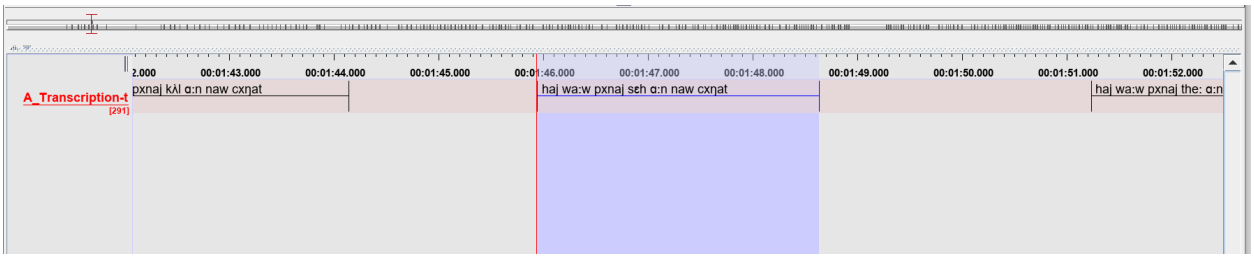


(8) Go back to ELAN and go to File > Import > CSV / Tab-delimited Text File. Import the .csv file you just created. Check off columns 1, 2, and 3 and match the columns to the correct values:

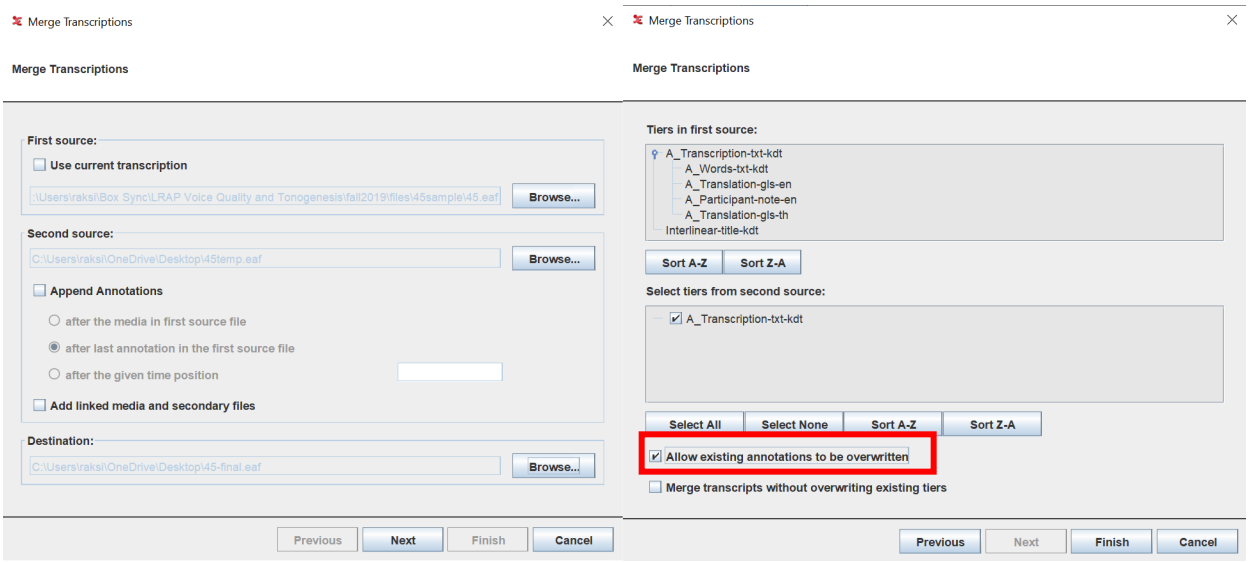
Begin Time, End Time, and Annotation. Make sure the first row of data is specified (it should be 2). Specify the delimiter as a comma. Press OK. If you are on a **Windows**, you may need to close all other ELAN files before doing this.



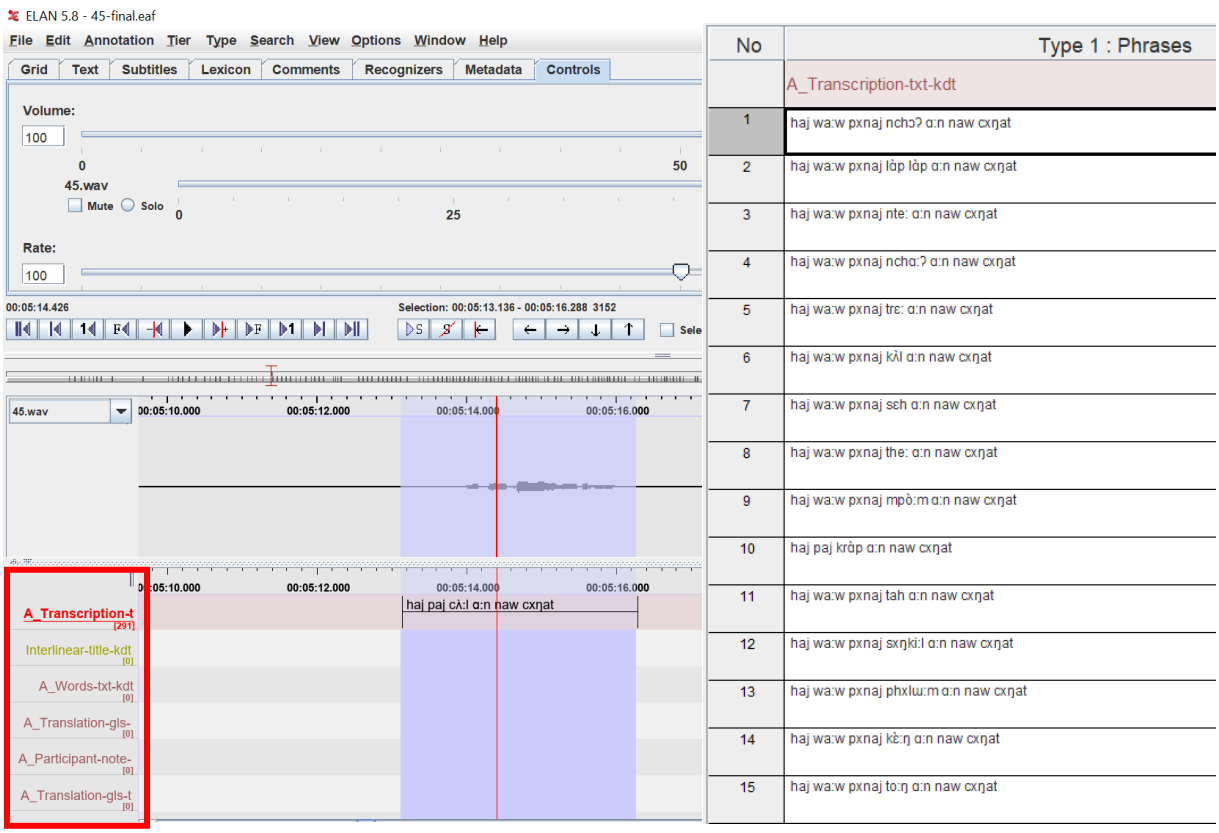
(9) Check to make sure it looks like the image before. There should only be one tier. It should have 290 annotations. If it looks good, Save As a .eaf file.



(10) Go to File > Merge Transcriptions and choose the *original transcription file* as the **First source** and then the .eaf file you just saved as the **Second source**. For **Destination**, make a name for the completed .eaf file (like 45-final.eaf, for example—the name doesn't matter much because eventually it will replace the original 45.eaf file). Press **Next** and then check off **Allow existing annotations to be overwritten**. Then press Finish.

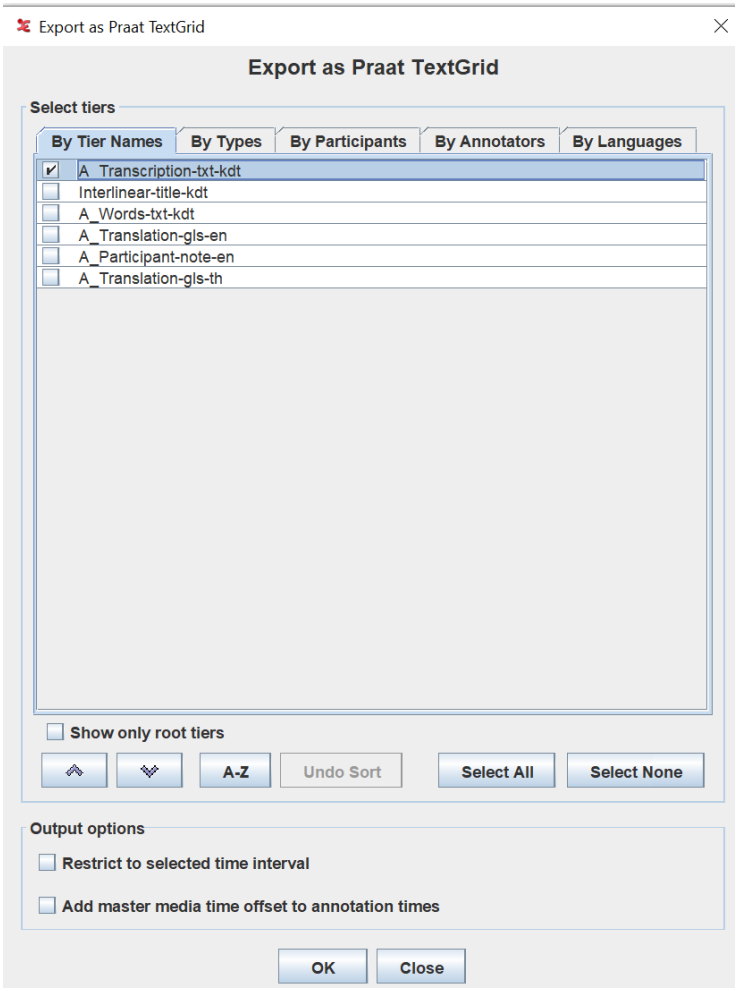


(11) Make sure you have the same 6 tiers as the original file (left image). Then check Transcription Mode to make sure you still have 290 segments and that they filled in correctly from the Excel file. If everything is good, feel free to replace the original speaker file (here, 45.eaf) with this file. Good job, you've finished the Annotation phase!



(B) Alignment

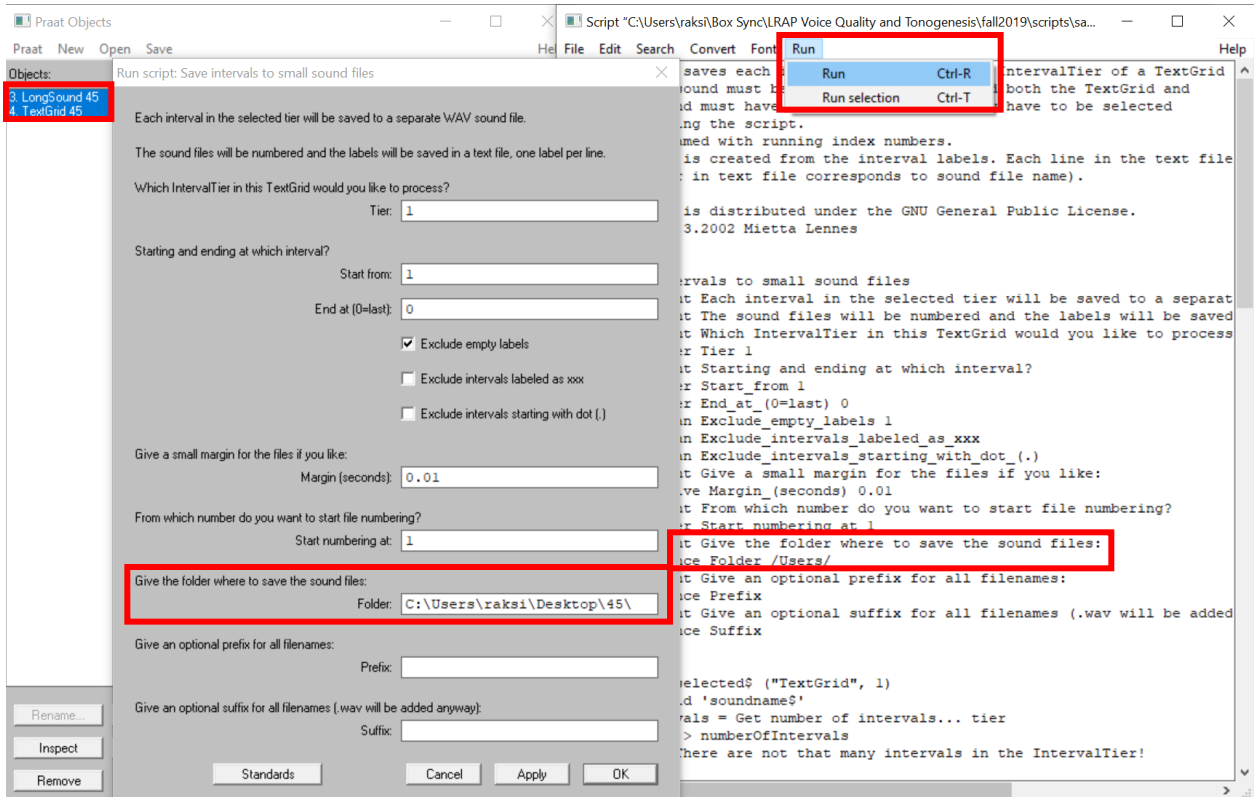
(1) In ELAN, go to File > Export as > Praat Text Grid. Check off only the transcription tier and press OK. Save it as your speaker number. Create a folder as well (wherever is convenient, I like to put it on my Desktop) with the speaker number as the name (ex. 45).



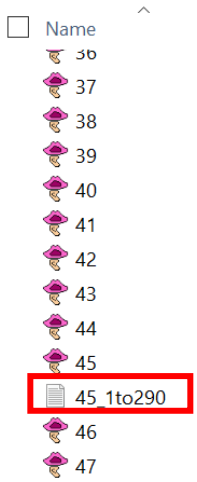
(2) Now we will do some Praat scripting. You may want to set .TextGrid and .praat files (and perhaps also .wav files) to open with Praat by default. To do this, you can right click any of these files and click "Get Info" (if Mac) or "Properties" (if Windows) and change the default program. Now, open up **save_intervals_to_wav_sound_files.praat** in Praat to separate the long sound file into individual sound files for each target word. After you open up the sound file,

Use **Open > Open long sound file** and choose the speaker .wav file. It is important that this is a **long sound** file. Now open up the accompanying TextGrid that you created in step (1) with **Open > Read from file** or by just double-clicking the TextGrid file. **Highlight** both the LongSound and the TextGrid files before running the script. To run the script, either choose "Run" under "Run, or press Command/Ctrl+R. Set the path to wherever you created the folder in (1), making sure you have a slash at the end, and press OK. There should be 290 files saved.

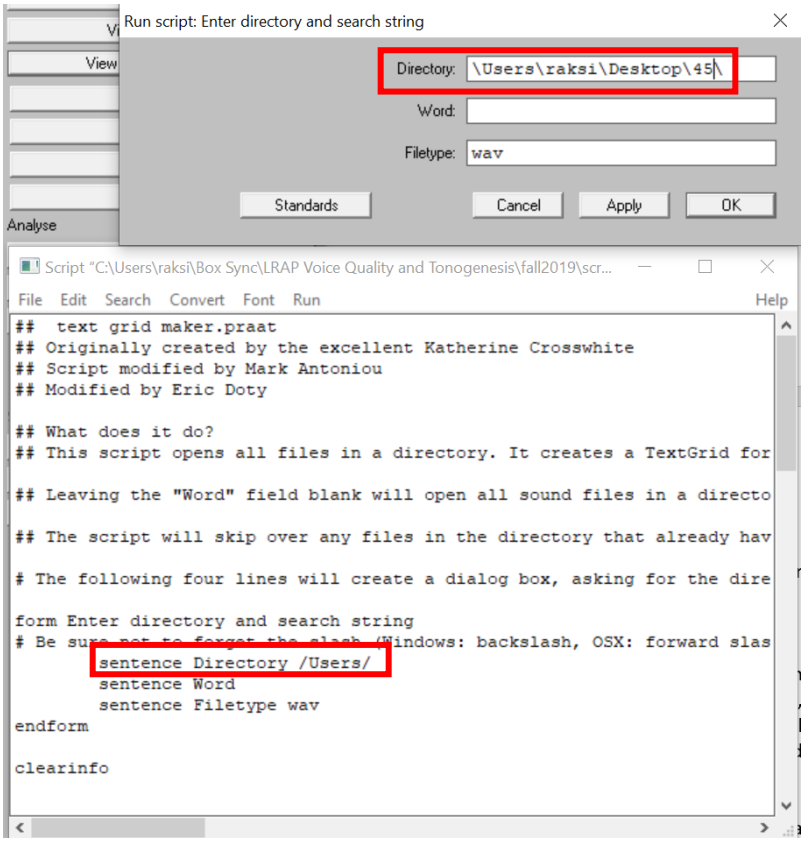
If you want to permanently change the folder where to save the sound files so that you don't have to retype every time, you can modify it directly in the script and then save the script.



(3) You will have a folder with a bunch of numbered files and an accompanying text file with all the transcriptions. There will also be a text file called [speaker number]_1to290.txt. Remember that this text file exists.



(4) Now use TextGridMaker.praat to create an empty text grid for each of these files. The tier should be called "transcription" automatically. The folder is the folder with the 290 files (remember the final slash). Once again, you can change the default folder.



(5) Now let's populate the empty TextGrids. First, neither MFA nor VoiceSauce play very well with Unicode, so we're going to translate the system into one without Unicode (ex. long vowels are z, breathiness is 1). Open up the jupyter notebook transcriptions.ipynb in the command line by first cd-ing to the scripts folder and then running:

jupyter-lab transcriptions.ipynb

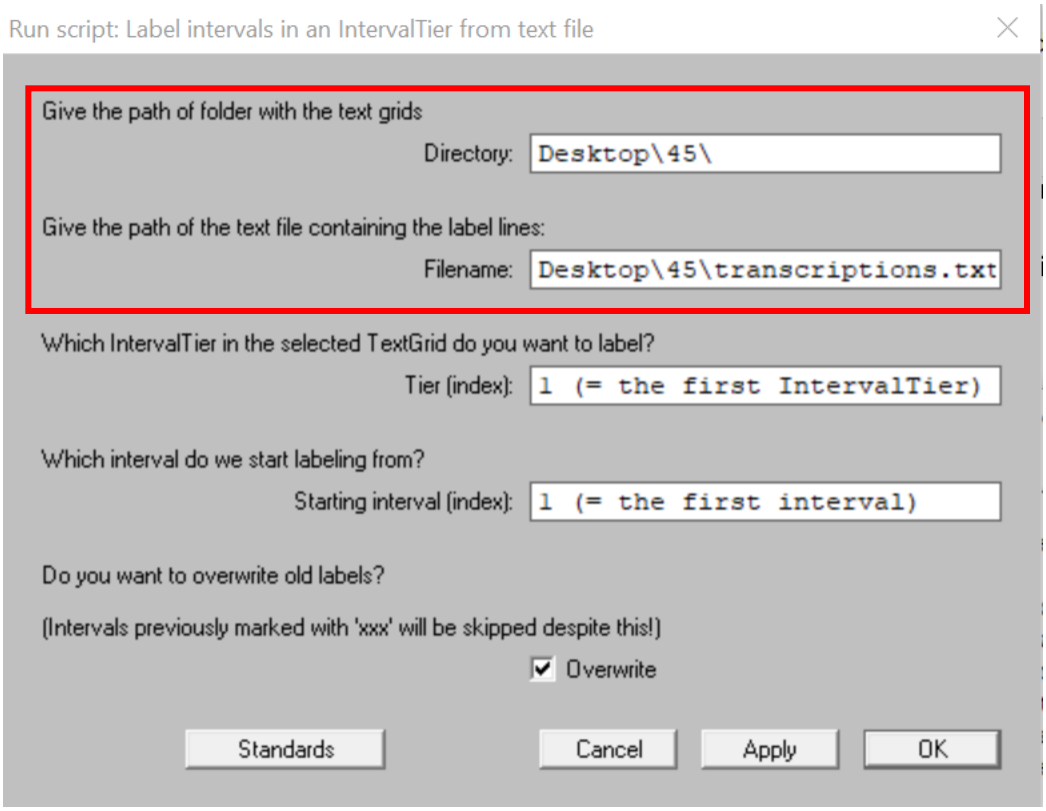
If you don't have jupyter-lab, you can either use it through the Berkeley Phonetics Machine (BPM) or you can set it up through anaconda. If you use the BPM, you can do the above command and skip down to the next paragraph. If you want to set it up on your computer, check out https://medium.com/better-programming/beginners-quick-guide-for-handling-issues-launching-jupyter-notebook-for-python-using-anaconda-8be3d57a209b . You must then also download audiolabel. To install audiolabel, you need to first install git (https://www.atlassian.com/git/tutorials/install-git). Once you have it installed, this command should work in your command line:

pip install git+https://github.com/rsprouse/audiolabel.git

You should move the folder you created to the scripts folder. In the first cell, change the variable **sp** to your speaker number. As a note, the **period** in a file name means **current folder**, i.e. the folder that the script is in. If all your packages are in order, the first cell in jupyter-lab should run smoothly. You can run it by just putting your cursor in the cell you want to run and then pressing Shift+Enter. Alternatively, you can do this all in the BPM, which already has jupyter-lab and audiolabel set up. Run all of Part 1. This will create a text file called transcriptions.txt in the scripts folder.

(6) Now we need to create a dictionary. Run Part 2. A dictionary.txt file should appear in the scripts folder.

(7) Move the transcriptions.txt and dictionary.txt files into the speaker folder with all the numbered files. We can now use the **label_from_text_file.praat** script, which will fill in the empty TextGrids from the transcriptions.txt file. Fill in the path to the .wav and .TextGrid files (this should end in a slash). The path to the transcription file should be the same path as to the .wav and .TextGrid files and should end in transcriptions.txt.

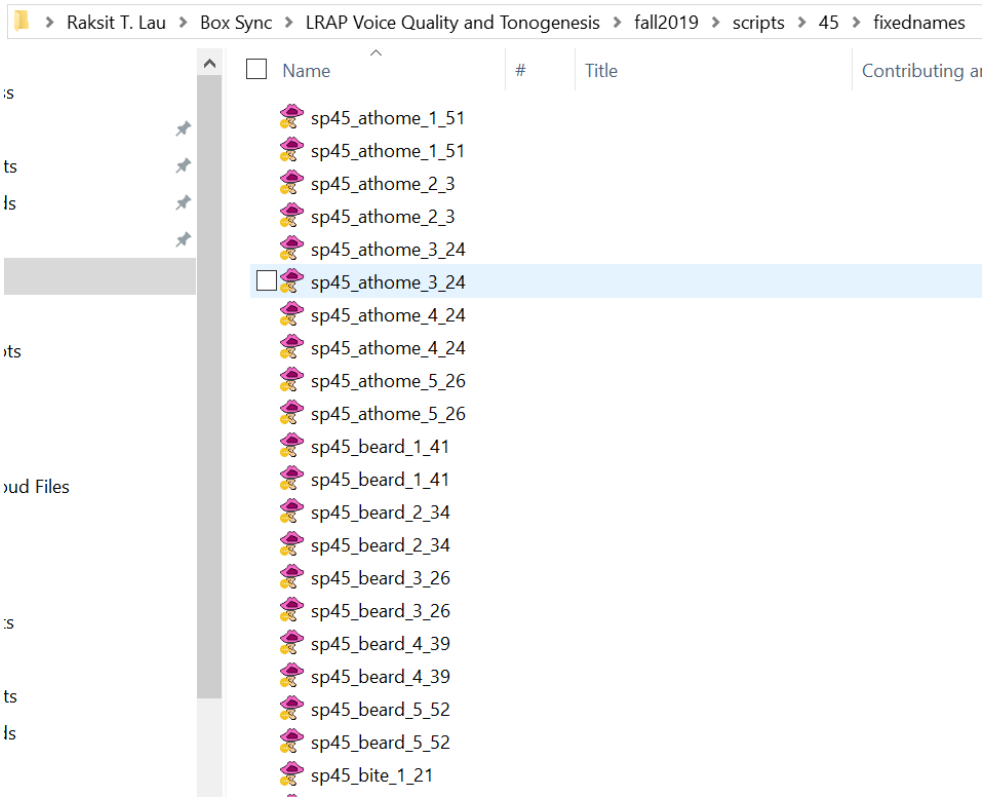


(8) Now we'll fix the file names. Make sure the clips.txt file you created in Annotation step (4) is in the speaker folder. Move the speaker folder to the scripts folder. Now, if you have a Windows, you will need to download Ubuntu (if you have a Mac, you do not need to) to run

the following command. Please follow the instructions here: https://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/ Now, open the command line, cd to the scripts folder, and type the following command, putting in your speaker number as the argument:

bash clip_names.sh [speaker number]

This will create a folder called fixednames that contains the .wav and .TextGrid files with the names fixed. They should be in the format [speaker number]_[word]_[round]_[number in round].



Check a random .wav file and its accompanying TextGrid (choose somewhere in the middle of the list) in Praat to make sure the files were named right. Check for what the word is in the speaker Excel file and listen to the audio and look at the TextGrid to see that it all matches. For example, if you listen to a file with the word “howl”, you should see “luz” in the and hear [lu:]. Remember that the TextGrid transcriptions are in the MFA-friendly transcription. You can look at the python dictionary in the second cell of the jupyter notebook to see what unfamiliar letters correspond to. For example, you will see that “z” corresponds to [:], which represents a long vowel.

```
# Dictionary for replacing
IPA_to_MFA = {
    # Diphthongs
    u'aj': '6',
    u'aw': 'v',
    u'a:w': 'v',
    # Vowels
    u'ɛ': '3',
    u'æ': '3',
    u'ɨ': 'y',
    u'ɯ': 'y',
    u'ə': '8',
    u'ɤ': '8',
    u'ʌ': '^',
    u'ɔ': '0',
    u'ɑ': '4',
    u':': 'z',
    # Breathiness
    u'ʾ':'1',
    # Consonants
    u'ŋ': '9',
    u'ɲ': '7',
    u'ʔ': 'q',
    # Other
    u'<': '',
    u'>': '',
    u'ˈ': '',
    u'ˠ': ''
}
```

Once you make sure everything is right, you can delete all the numbered files. **BEFORE YOU DO THIS, move the [speaker number]_1to290.txt file to the original folder that I assigned you**. Move clips.txt and transcriptions.txt as well to this folder. Now, you can move the correctly named files up to where the numbered files were and delete the fixednames folder.

(9) Now let's align! If you haven't set up VirtualBox and the BPM, please follow the instructions at this link: http://linguistics.berkeley.edu/plab/guestwiki/index.php?title=Berkeley_Phonetics_Machine . In VirtualBox, you should add to Shared Folders whatever folder on your computer contains the files you will be working with. To do that, follow the instructions in the "Shared folders" section in the link. Open up the BPM and enter the following in the Terminal in the BPM after first cd-ing to the folder with your corpus and dictionary first (in the BPM, you can drag the folder to the Terminal after typing cd):

    # To validate:
    /opt/montreal-forced-aligner/bin/mfa_validate_dataset [CORPUS DIRECTORY] [DICTIONARY FILE]

    # To align:
    /opt/montreal-forced-aligner/bin/mfa_align [CORPUS DIRECTORY] [DICTIONARY FILE] [LANGUAGE] [OUTPUT DIRECTORY]

- Corpus directory: the directory with all .wav and .TextGrid files to be aligned
- Dictionary file: path to where the dictionary.txt is
- Language: the language the aligner is trained on--we will use English
- Output directory: the path (and name) of the folder you want the output to end up in (it cannot be the same as the corpus directory). You do not have to create this folder beforehand. The aligner will create it.

A period (.) represents the "current directory"

/opt/montreal-forced-aligner/bin/mfa_validate_dataset . ./dictionary.txt

/opt/montreal-forced-aligner/bin/mfa_align . ./dictionary.txt english ./output

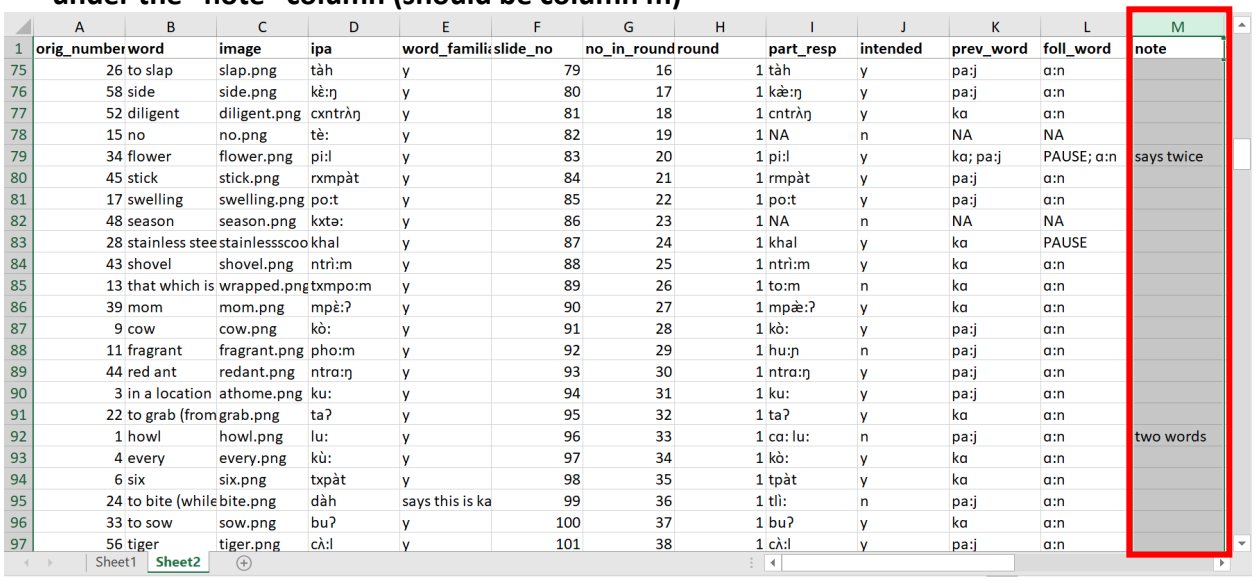
You should see two tiers: transcription - words and transcription - phones in the files in the output folder.

Often problems crop up here at the validation stage. It may tell you that there are missing words and you can check oovs_found.txt and oovs_utterance.txt in the path it tells you to figure out what TextGrid is causing the problem. It is very likely you will have some unalignable files (but it should be suspicious if almost everything is unalignable).

Great, you are done with the Alignment phase!

(C) Realignment

(1) **If you come across any issues with realignment, especially in terms of not being sure about where to make a boundary (sometimes the spectrogram and wave won't agree, or it will be very ambiguous), please make a note in Sheet 2 in the speaker Excel file under the "note" column (should be column m)**



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	orig_number	word	image	ipa	word_famili	slide_no	no_in_round	round	part_resp	intended	prev_word	foll_word	note
75	26	to slap	slap.png	tàh	y	79	16	1	tàh	y	pa:j	ɑ:n	
76	58	side	side.png	kɛ̀:ŋ	y	80	17	1	kæ̀:ŋ	y	pa:j	ɑ:n	
77	52	diligent	diligent.png	cɤntrʌ̀ŋ	y	81	18	1	cntrʌ̀ŋ	y	kɑ	ɑ:n	
78	15	no	no.png	tè:	y	82	19	1	NA	n	NA	NA	
79	34	flower	flower.png	pi:l	y	83	20	1	pi:l	y	kɑ; pa:j	PAUSE; ɑ:n	says twice
80	45	stick	stick.png	rxmpàt	y	84	21	1	rmpàt	y	pa:j	ɑ:n	
81	17	swelling	swelling.png	po:t	y	85	22	1	po:t	y	pa:j	ɑ:n	
82	48	season	season.png	kxtə:	y	86	23	1	NA	n	NA	NA	
83	28	stainless stee	stainlessscoo	khal	y	87	24	1	khal	y	kɑ	PAUSE	
84	43	shovel	shovel.png	ntrì:m	y	88	25	1	ntrì:m	y	kɑ	ɑ:n	
85	13	that which is	wrapped.png	txmpo:m	y	89	26	1	to:m	n	kɑ	ɑ:n	
86	39	mom	mom.png	mpɛ̀:ʔ	y	90	27	1	mpæ̀:ʔ	y	kɑ	ɑ:n	
87	9	cow	cow.png	kò:	y	91	28	1	kò:	y	pa:j	ɑ:n	
88	11	fragrant	fragrant.png	pho:m	y	92	29	1	hu:ŋ	n	pa:j	ɑ:n	
89	44	red ant	redant.png	ntrɑ:ŋ	y	93	30	1	ntrɑ:ŋ	y	pa:j	ɑ:n	
90	3	in a location	athome.png	ku:	y	94	31	1	ku:	y	pa:j	ɑ:n	
91	22	to grab (from	grab.png	taʔ	y	95	32	1	taʔ	y	kɑ	ɑ:n	
92	1	howl	howl.png	lu:	y	96	33	1	cɑ: lu:	n	pa:j	ɑ:n	two words
93	4	every	every.png	kù:	y	97	34	1	kò:	y	kɑ	ɑ:n	
94	6	six	six.png	txpàt	y	98	35	1	tpàt	y	kɑ	ɑ:n	
95	24	to bite (while	bite.png	dàh	says this is ka	99	36	1	tlì:	n	pa:j	ɑ:n	
96	33	to sow	sow.png	buʔ	y	100	37	1	buʔ	y	kɑ	ɑ:n	
97	56	tiger	tiger.png	cʌ̀:l	y	101	38	1	cʌ̀:l	y	pa:j	ɑ:n	

Sheet1 Sheet2

(2) We will be focusing on the following modal-breathy minimal pairs:

athome [ku:] – every [kù:]
sun [puːʔ] – beard [pùːʔ]
coconut [toːŋ] – male [tòːŋ]
divorce [tah] – slap [tàh]
return [lɑp] – dusk [lɑ̀p] **(note below)**
flower [piːl] – spin [pìːl]
fragrant [phoː m] – just [mpòː m]
grab [taʔ] – placeunder [tàʔ]
hide [nchuː n] – send [cùː n]
jar [theː] – no [tèː] – tell [nteː] (note below)
howl [luː] – thigh [lùː]
old [tiː] – tall [tìː]
waist [ŋkɛ̀ː ŋ] – side [kɛ̀ː ŋ]
west [txpat] – six [txpàt]
swelling [poːt] – toomuch [pòːt]

For each minimal pair, check each .wav file and its accompanying TextGrid to see if the word that is said corresponds to the actual intended target word. For example, if we take the pair 'howl' and 'thigh', we would look for all 5 howl files and all 5 thigh files. For howl, we would listen for whether [luː] was said, and for thigh, whether [lùː] was said. If the target word was not said OR if it is in a compound, we will exclude that file. Let's say for this pair, the words said were (the number after the gloss is the round number):

howl_1: luː	thigh_1: lùː
howl_2: krɑh	thigh_2: lùː
howl_3: luː	thigh_3: kòːn lùː
howl_4: luː	thigh_4: kòːn lùː
howl_5: krah	thigh_5: kòːn lùː

We would count howl_1, 3, and 4; and thigh_1 and 2. We exclude krah because it was not the target word and kòːn lùː because it is a compound.

Common words you may hear in place of minimal pair words:

athome	[kuː]	
every	[kùː]	krɑ̀p 'each', tùk 'suffering', txŋaj 'day', kù txŋaj ‘every day’
sun	[puːʔ]	kxtaw 'hot', ntiàŋ 'sun', txŋaj 'day', mxnaŋ ‘sunlight’
beard	[pùːʔ]	ɲah ‘curly’
coconut	[toːŋ]	plaj toːŋ ‘coconut fruit’
male	[tòːŋ]	tɑː tòːŋ 'male', tòːl 'male'
divorce	[tah]	riàŋ 'divorce' (Thai)
slap	[tàh]	
return	[lɑp]	

dusk	[làp]	làp làp ‘dusk’, txbɯ: 'evening'
flower	[piːl]	kʌ̀l piːl 'flower', piːl aluàŋ ‘flower’
spin	[pìːl]	mùːl 'to spin', pàn 'to spin', muàn 'to spin'
fragrant	[phoːm]	huːɲ 'to sniff'
just	[mpòːm]	mpòːm juːl 'just got up', sxʔaːp 'to yawn', adɯh 'before'
grab	[taʔ]	
placeunder	[tàʔ]	rɔ̀ːŋ ‘to place under’ (Thai), tàŋ ‘to place under’, tìːl ‘to place under’
hide	[nchuːn]	abuŋ ‘spoon’
send	[cùːn]	
jar	[theː]	sxluːɲ 'jar'
no	[tèː]	phiʔ aj 'is not', tèːtèː 'no'
tell	[nteː]	sxnteː ‘tell’
howl	[luː]	hɔːn (the Thai word), krɑh 'to bark'
thigh	[lùː]	kòːn lùː (compound word for 'thigh'), jɯːŋ 'foot', cxŋkàŋ 'shin'
old	[tiː]	
tall	[tìː]	
waist	[ŋkɛ̀ːŋ]	
side	[kɛ̀ːŋ]	kɛ̀ːm 'side'
west	[txpat]	ntiàŋpat 'west', ntiàŋ ntruh 'sunset
six	[txpàt]	hɑk 'to fall', ntrùh 'to fall'
swelling	[poːt]	ploːh 'bruise'
toomuch	[pòːt]	daːl 'a lot', daːl pòːt 'too much'

You will notice that some of the target words don't quite look like minimal pairs because there is prenasalization (these are bolded); for example, the triplet [nteː] 'to say', [tèː] 'no', and [theː] 'jar'. Some speakers (especially younger ones) do **not** have prenasalization, making these minimally different (i.e. [nteː] becomes [teː]). Listen for whether a speaker has prenasalization on these specific words and also look for evidence in the spectrogram. Note that some speakers may prenasalize the word “side”.

Please fill in the template-prenasalization.xlsx file and save it replacing “template” with your speaker number (i.e. 45-prenasalization.xlsx). Fill in how many tokens each speaker has for each word that are prenasalized, not prenasalized, or where they say another word (or no word). These three categories should add up to 5 for each word.

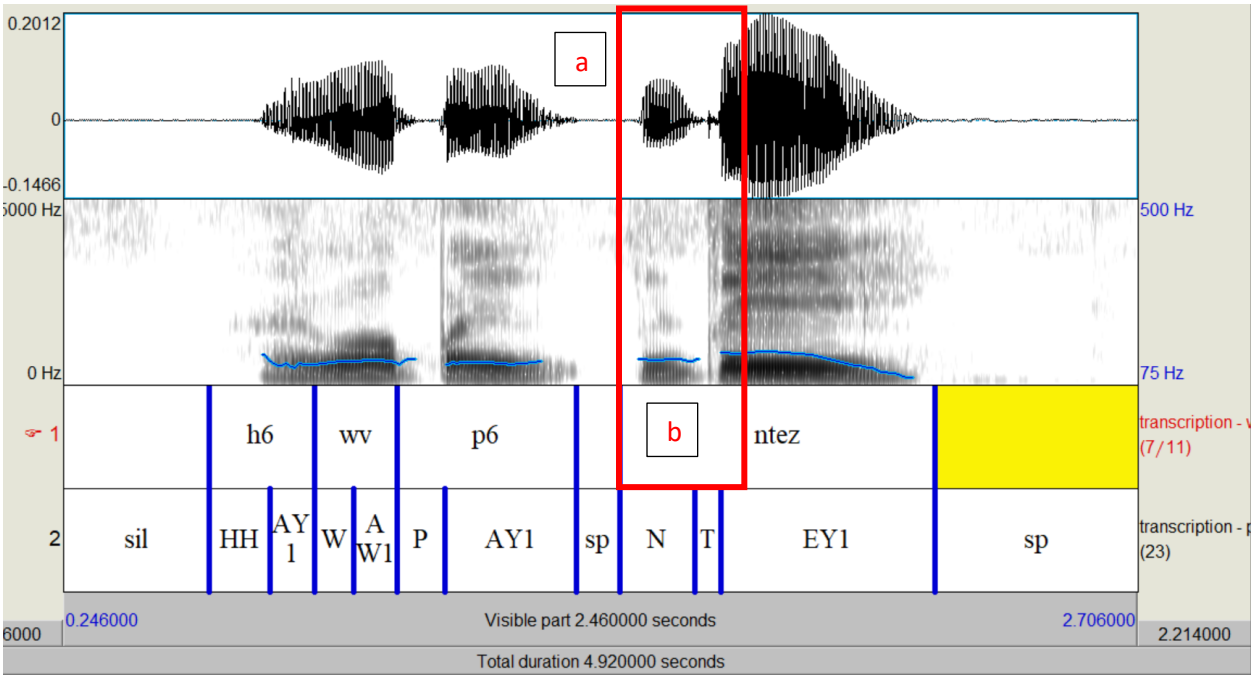
The images below offer some examples (you can find these examples in the “otherexamples” folder in). The letters in the following list correspond to the letters in the images. In the first image, the speaker has clear prenasalization:

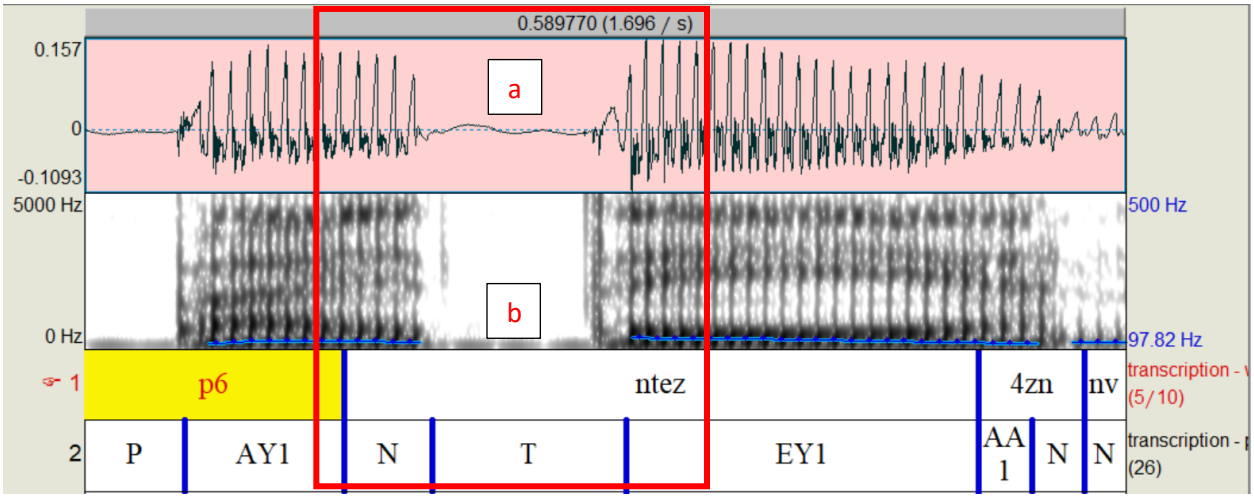
(a) Note the waveform here is not as large as in the [eː] but is still existent and periodic. This suggests the existence of the nasal

(b) Notice there is also spectral energy at low frequencies and some faint bands at higher energies that match the following [e:] roughly. All this comes before the silent white part, with a heavily reduced waveform, that is [t].

Now look at the second image. This speaker does not have prenasalization. While it looks like the forced aligner found an [n], notice that this is actually the sequence [aj] in [paj]. You can listen to see that that is not an [n] too.

(a) Notice that there is no clear weaker periodic waveform than the [e:] part, unlike in the top photo. Instead, the waveform is almost 0, suggesting that this is just a silence (i.e. a stop).
(b) Don't be fooled that it looks like there is low spectral energy. The lack of *any* energy at higher frequencies—notice that it is fully white, unlike in the top picture—should be suspicious. The low energy may be something else, as you can see it is in the stop [p] at the left end of the window as well.





For the words with potential prenasalization, you can count in the minimal pairs if they are *not* prenasalized (i.e. [nte:] and [tè:] are not a minimal pair, but [te:] and [tè:] are. Note that you should check every word because some speakers vary between prenasalizing and not.

Most speakers who actually say the intended target for 'dusk' will say [làp làp]. However, *some* speakers will say just one syllable [làp]. If they say one syllable, then it is a minimal pair. Make sure to delete the second syllable in the WORD tier if this is the case.

As long as there are at least 2 valid (i.e. what is produced is an actual minimal pair) files for each word of the pair we will then set these files aside for realigning. If only thigh_1 was valid, for example, then we would not realign the howl-thigh pair *at all*.

For the minimal triplet [the:] 'jar' vs. [nte:] 'tell' vs. [tè:] 'no', you can include them in analysis if there are enough tokens for at least two of the words.

(3) Create a folder in the speaker folder called "realigned" with the speaker number (ex. 14-realigned). If you end up realigning a TextGrid (you almost certainly will), save the **new realigned** .TextGrid file in this folder and move the .wav file into it as well.

(4) The things you will do when realigning are:

a. Realign both the left and right boundaries of the main vowel in the target word (ex. [u:] in [lu:])

b. Create a VOT segment (if the last onset consonant is a stop) in between the last onset consonant and the vowel (i.e. between [t] and [a] in [tah] or between [p] and [a] in [tpat]). VOT stands for Voice Onset Time, the amount of time between the stop

release and the onset of voicing (i.e. periodic noise). **If the stop is aspirated (i.e. in [pho:m], you should change the forced alignment "HH" label to "VOT".**

(4) The most important things to look for when realigning are a clear change in the waveform *and* in the spectrogram. Sometimes, the changes will be unfortunately ambiguous and you will have to make a judgment call (and/or ask me). Some general principles:

a. If the vowel starts with a cycle moving *away* from 0 dB, make sure it ends with a cycle moving *toward* 0 dB.
b. Place boundaries at zero crossings if you can. For the burst at the beginning of VOT, you may have to just choose a point by hand (corresponding to a large change in waveform and spectral energy) if the nearest 0 crossing is too far away.
c. Bursts and transitions from the consonant into the vowel are characterized by large and noticeable changes in waveform and spectral energy.
d. If the end of a vowel is transitioning into another vowel (often it will, because of the following [ɑːn], try to choose a point before the formants transition.
e. If the vowel is followed by a glottal stop, you will notice the cycle getting stretched out (i.e. lower frequency). Make the boundary between the vowel and glottal stop (which is approximated by the ARPAbet as "T") just before the cycles get extremely stretched out.
f. If the vowel is followed by no coda and silence, it can be hard to pick the endpoint. Look at where the spectral energy looks like it's really dying out and when the waveform noticeably changes
g. Make sure boundaries between phonemes go on Tier 2—shift the Tier 1 boundaries only if the ends of the word shift.

Useful shortcuts (I will write Command to mean the Apple Key—it is Ctrl if you have a windows)
Cmd + O: Zoom out
Cmd + I: Zoom in
Cmd + N: Zoom to selection
Cmd + 0: Move cursor to nearest zero crossing
(if you have highlighted an interval and not a point):
    Cmd + , : Move left boundary to nearest zero crossing
    Cmd + . : Move right boundary to nearest zero crossing
Alt + Backspace: Delete a selected boundary

Another useful trick is that if you select a position (say, a zero crossing), and then hold Shift while dragging a boundary near the red line you selected, it will "snap" to the position if it is close enough—this is useful so that you don't have to create boundaries and delete!

If you make a mistake, it is better to manually click **Edit > Undo …** For some reason, Cmd + z sometimes doesn't work how you expect it to.

Example Realignment

I will use examples from the **demonstrations > aligning > 14-examples** folder. As a tip, it might be easier to see differences in formants if you go to **Spectrum > Spectrogram settings** and set the dynamic range to 50 dB.

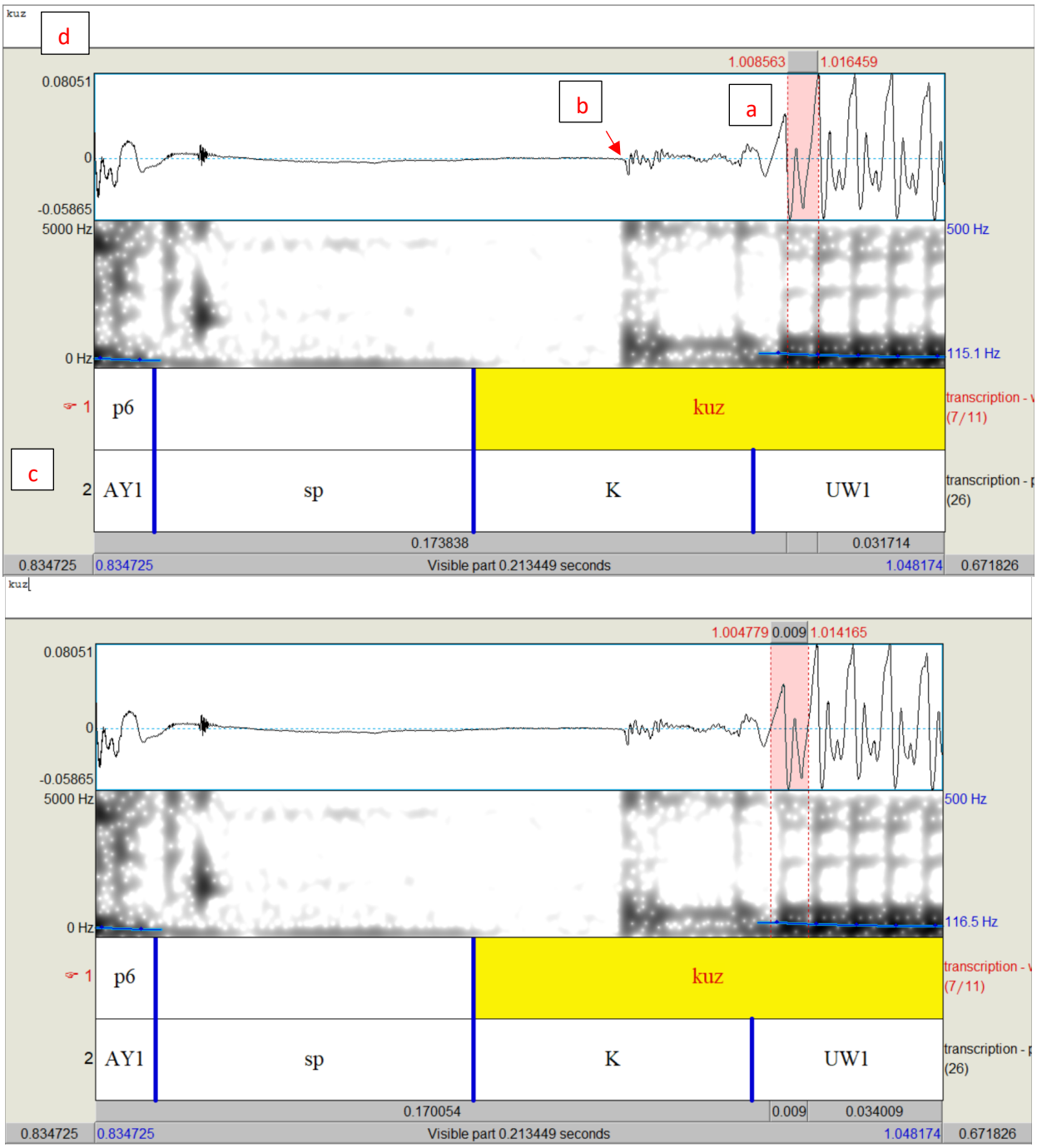
The image below is from sp14_athome_1_41. Let's first find the left boundary of the vowel [u:] (UW1) (the transcription the Montreal Forced Aligner uses is called ARPAbet—you can check Part 2 in the jupyter transcriptions notebook to see how our MFA-friendly transcription maps to ARPAbet).

a. Just after (a), notice that the waveform begins to look like very much like what it looks like for the duration of the vowel. This is where the vowel begins. There is also evidence in the spectrogram—notice that the formants (the dark areas) begin to show up here. So this is the beginning of [u:].

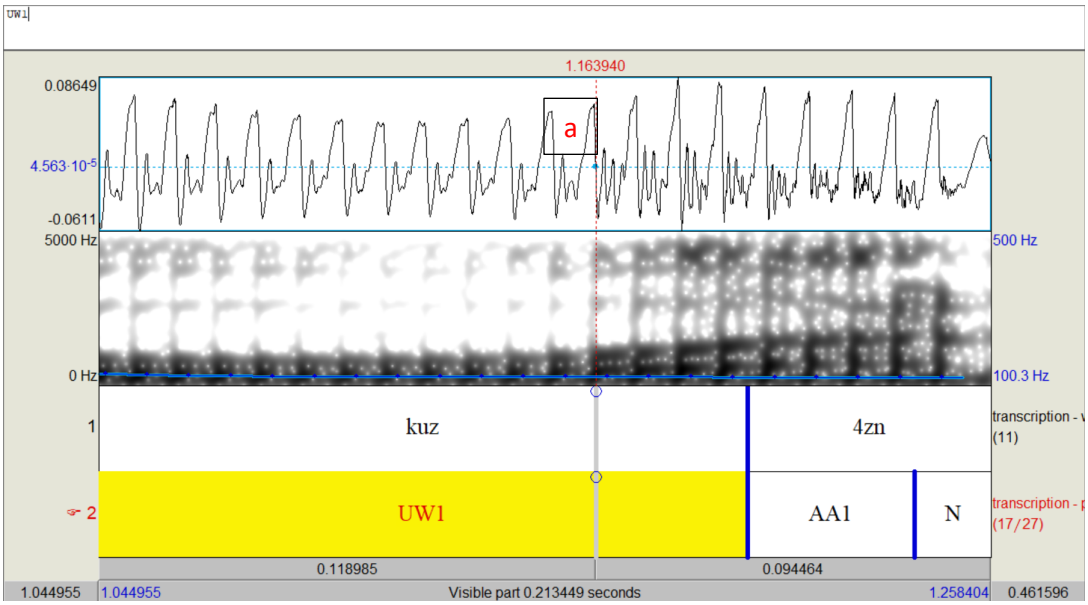
   First, identify what a cycle looks like. Here, I have highlighted the first cycle for you. Note that a cycle can start with either a fall into the negative amplitude range (as in the top picture) or a rise into the positive amplitude range (as in the bottom picture). You will have to make a judgment call where the cycle starts. Both are valid cycles, but notice that the boundary in the spectrogram makes more sense in the top picture, so we will go with that.

b. Now we can mark the left boundary of the VOT. The left boundary of VOT (the "burst"--i.e. the release of the stop) is generally pretty clear, as the waveform will go from 0 to aperiodic vibration, as seen at point b. The spectral energy also goes from 0 to the beginning of some formants showing. You should try to place the boundary at a zero crossing near the burst, but if the nearest zero crossing is far away, just make the boundary by hand.

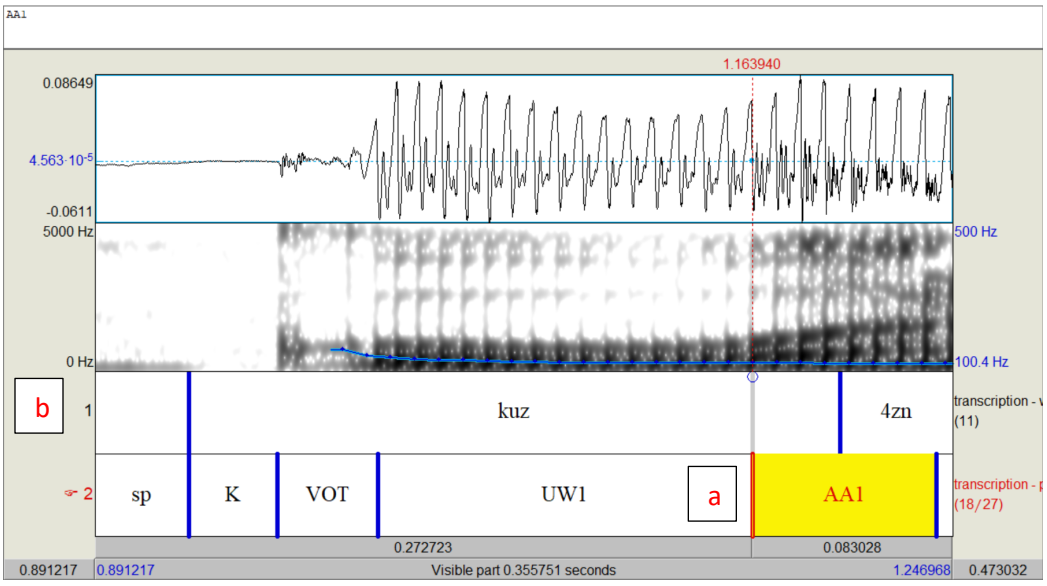
c. When you select a point, you can press **Enter** to create a new boundary. Note that it will create a boundary **on the tier being pointed to by the red hand.** If you want to change which tier you add a boundary to, you must **click the grey area where the hand is to the left of the tier you want to select (where c is)** before pressing Enter. Note that when you create a boundary, the labels in the segments may get moved around. You may have to cut and paste the labels to get them right again. When you select an interval, the label in it will appear at the top (see point d below). You can manipulate the label here. To delete a boundary, select it and press Alt+Backspace.



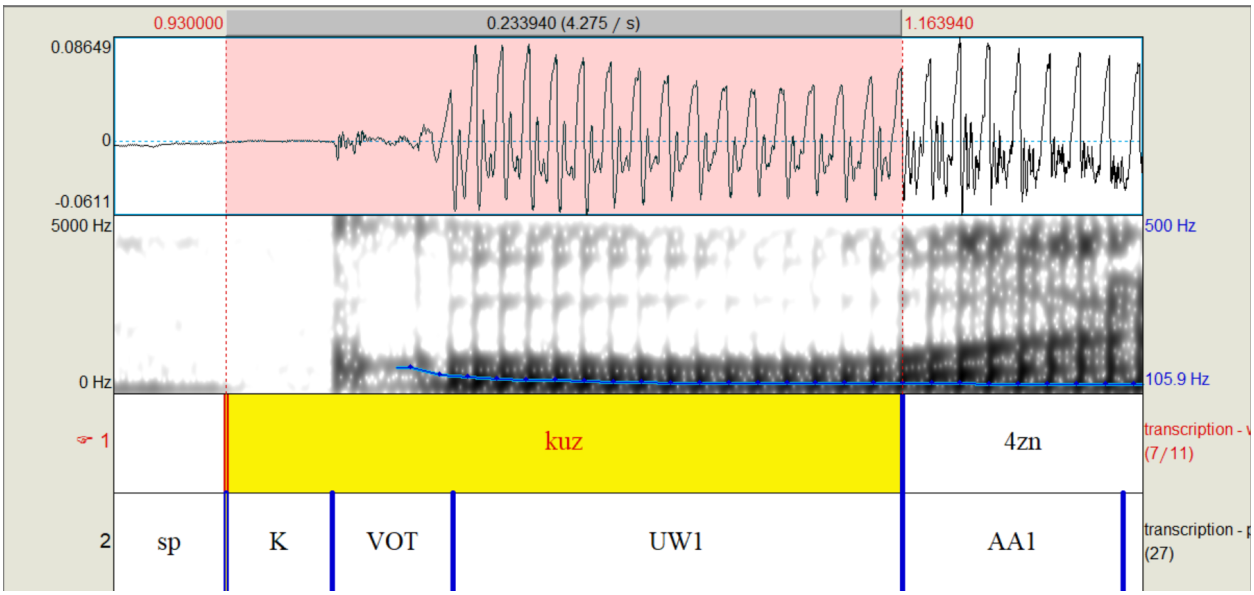
Now let's look at the right boundary of the [u:] vowel. Where the cursor is near (a), you will notice that the waveform changes a lot (the second crest of the wave is sharply higher and there is a third crest too). If you look at the spectrogram, you will also see that the second formant is beginning to increase here. Thus, this is the boundary we will choose.



Note that Tier 1 boundaries should only be shifted if the ends don't match the Tier 2 boundaries. In the image below, notice that shifting the force-aligned right boundary of UW1 left leads to an inconsistency with the right boundary of kuz in Tier 1. To fix this, click the boundary to the right of point a and then click point b to select Tier 1. **Then** click Enter. You can delete the old boundary.

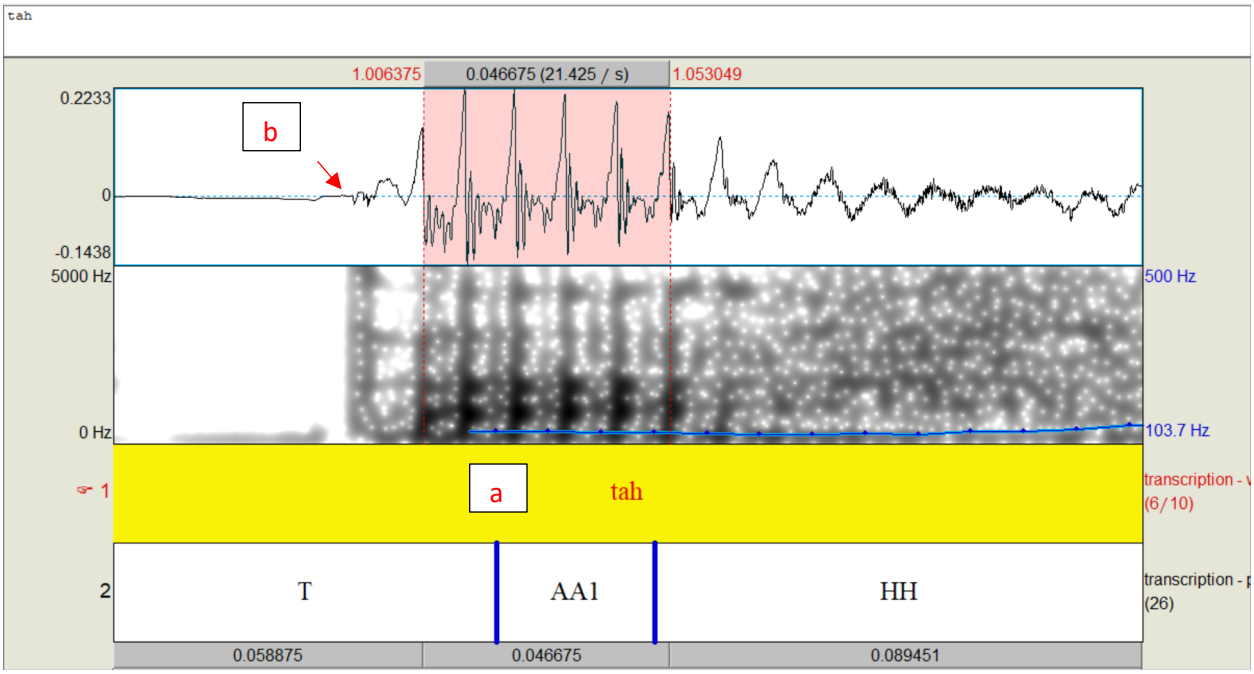


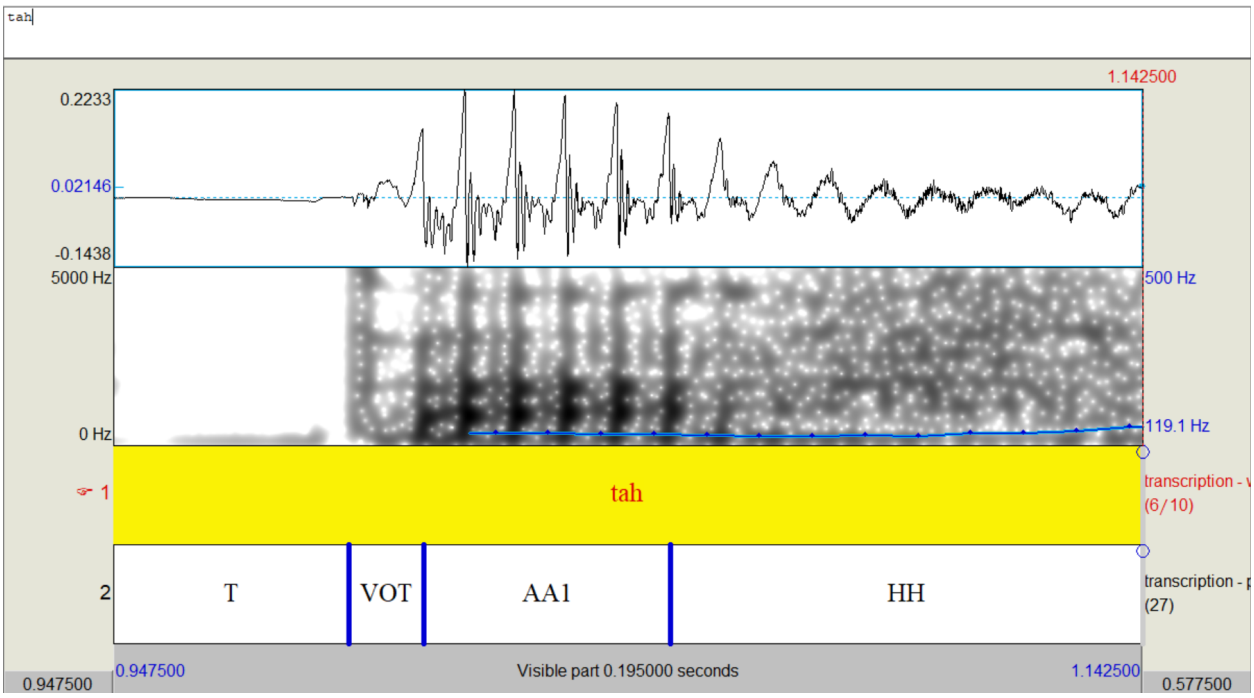
This is the final result of the realignment:



sp14_divorce_2_36

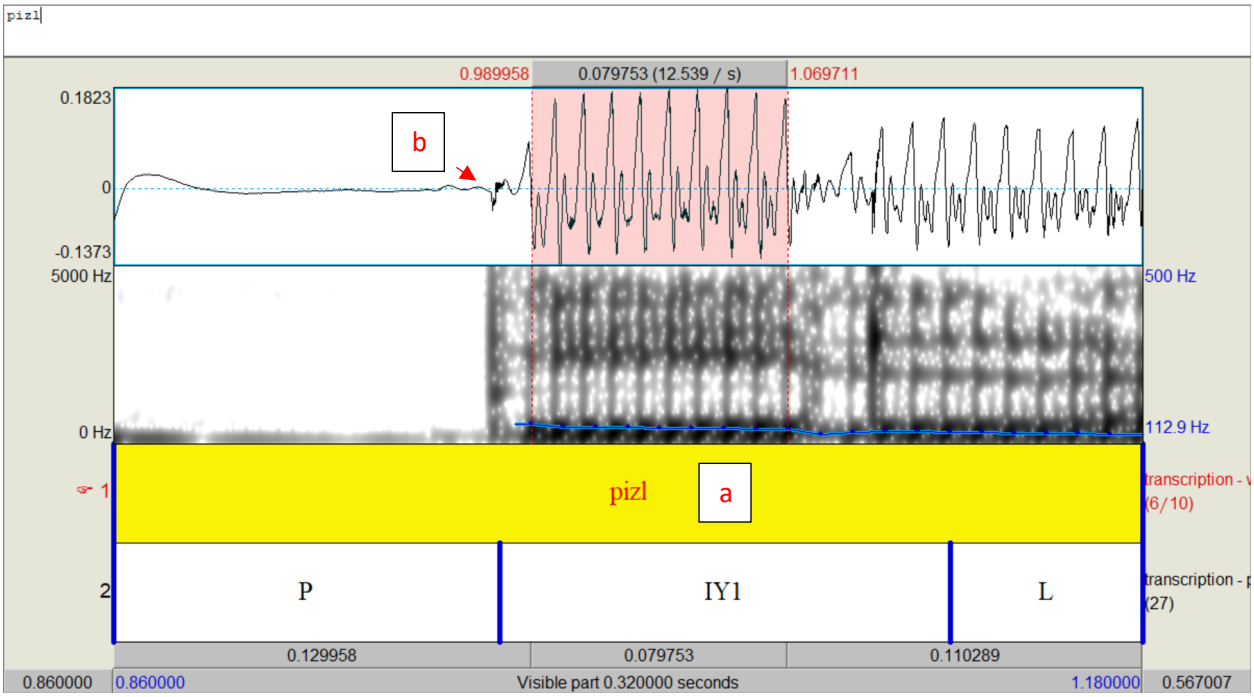
The selected red portion (above a) is where we will delineate the vowel boundaries. Note that the left end is where we see the beginning of the periodic waveform and also where the dark vowel formants begin. The right end is where the periodic waveform ends and also where the vowel formants begin to become lighter and more scattered (this is the aperiodic noise of the fricative [h]). The left end of the VOT is where the arrow from b is pointing—it is where the waveform stops being flat and also where spectral energy begins. The resultant realignment is below.

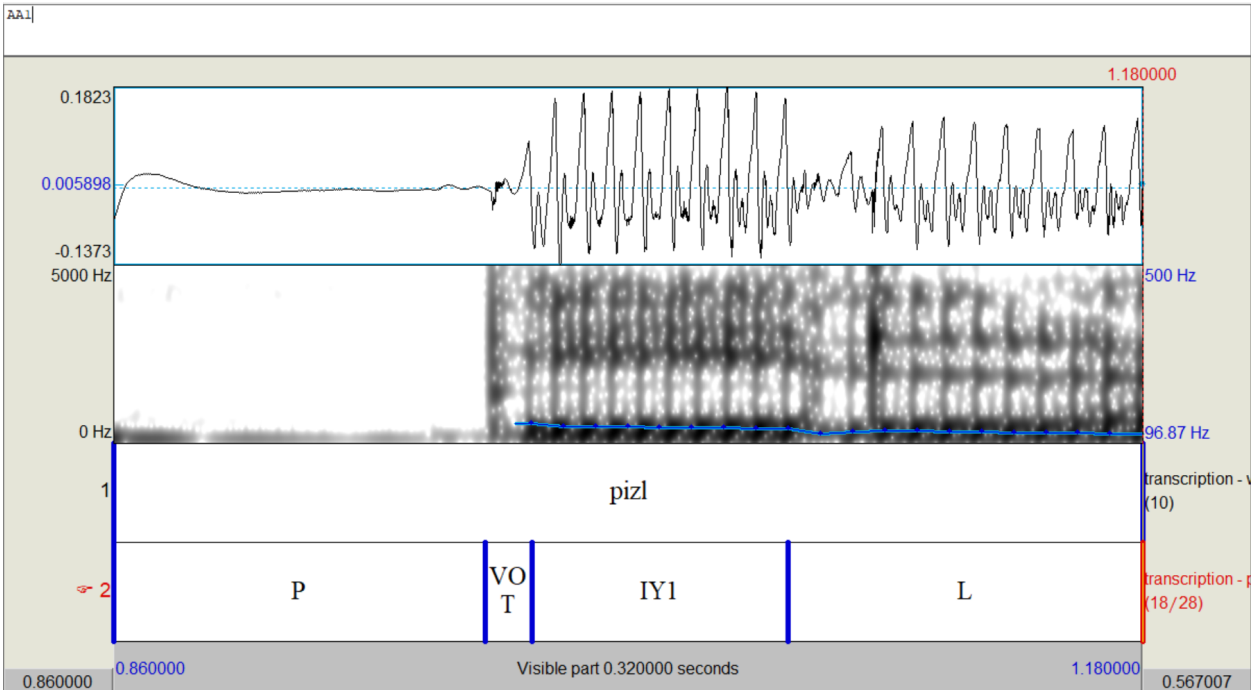




sp14_flower_1_24

Notice that the boundary for [l] is way off from where we expect it. This happens a lot with [l] (my hunch is that it is because this MFA is trained on English [l], which is often velar, but Kuy [l] is not velar at all). Once again, the left end of the vowel is noticeable as it is the beginning of the regular cycles (which each have two crests) and where the vowel formants begin. The right end is also marked by a clear change in the waveform and some faintness of the spectral energy. Where b is pointing is the beginning of the VOT as it changes from silence there.

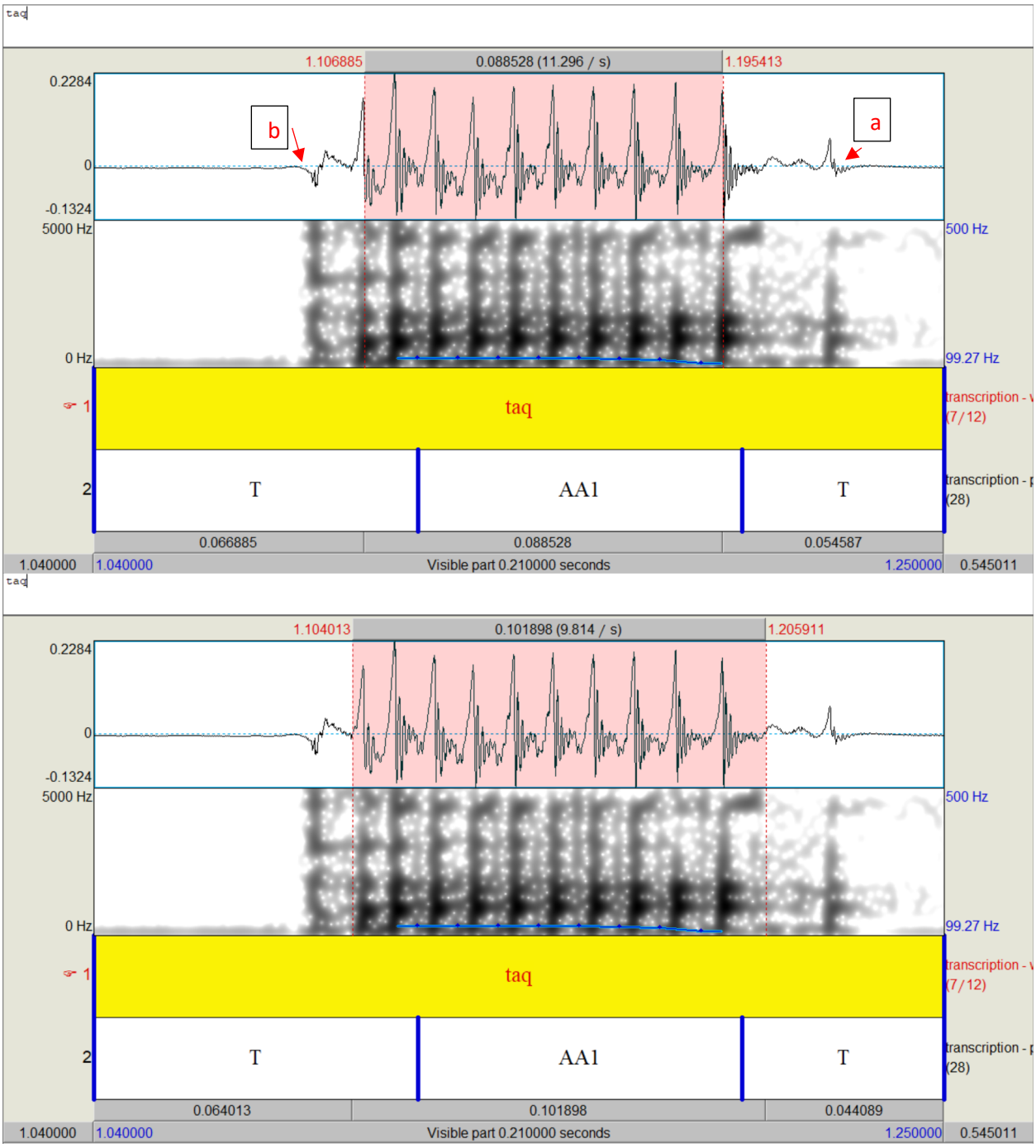




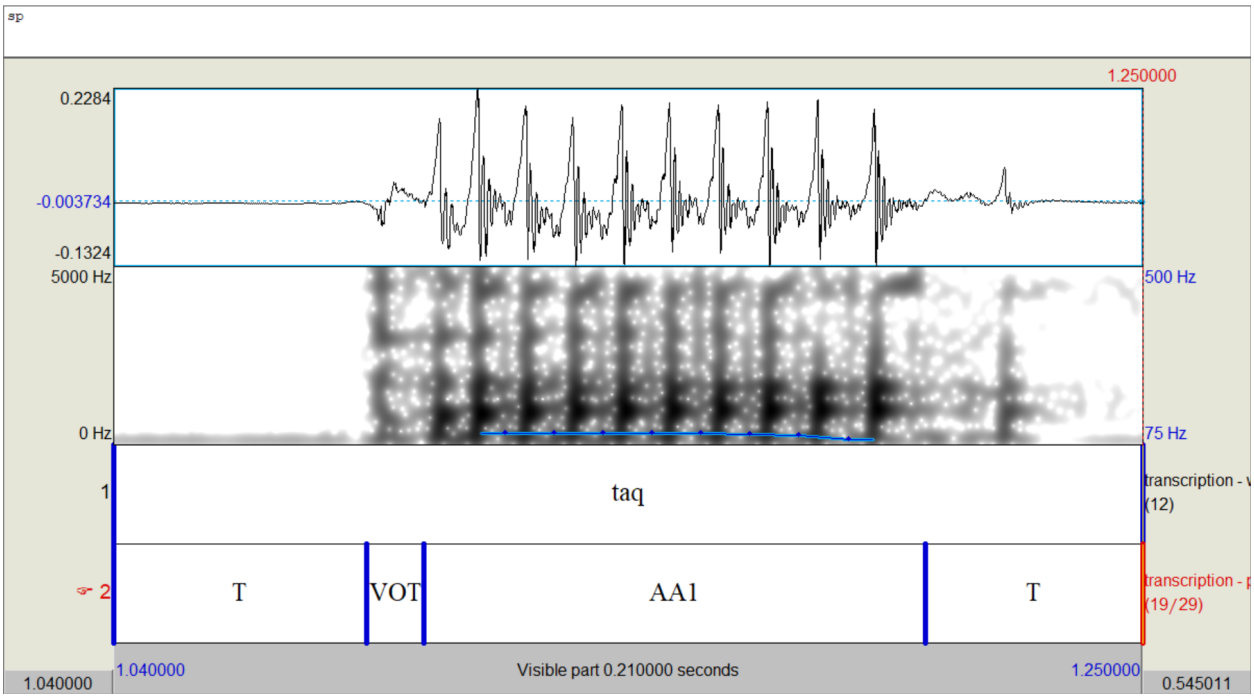
sp14_grab_1_26

The highlighted region is where the vowel is. Notice the waveform from the end of the vowel to point a and how it looks kind of like the other cycles, but stretched out. This is characteristic of glottal stops (represented by <q> in our MFA-friendly transcription and approximated by <T> in ARPAbet). Glottal stops severely reduce pitch and frequency, leading a longer period (and thus, it will be more "stretched out"). In some of the files, you may see several stretched out waveforms. Make the boundary before the stretching out becomes noticeable.

The lower image is another alternate segmentation of the vowel duration. Notice that it matches the spectral energy changes a bit better. Either way would be acceptable to me.

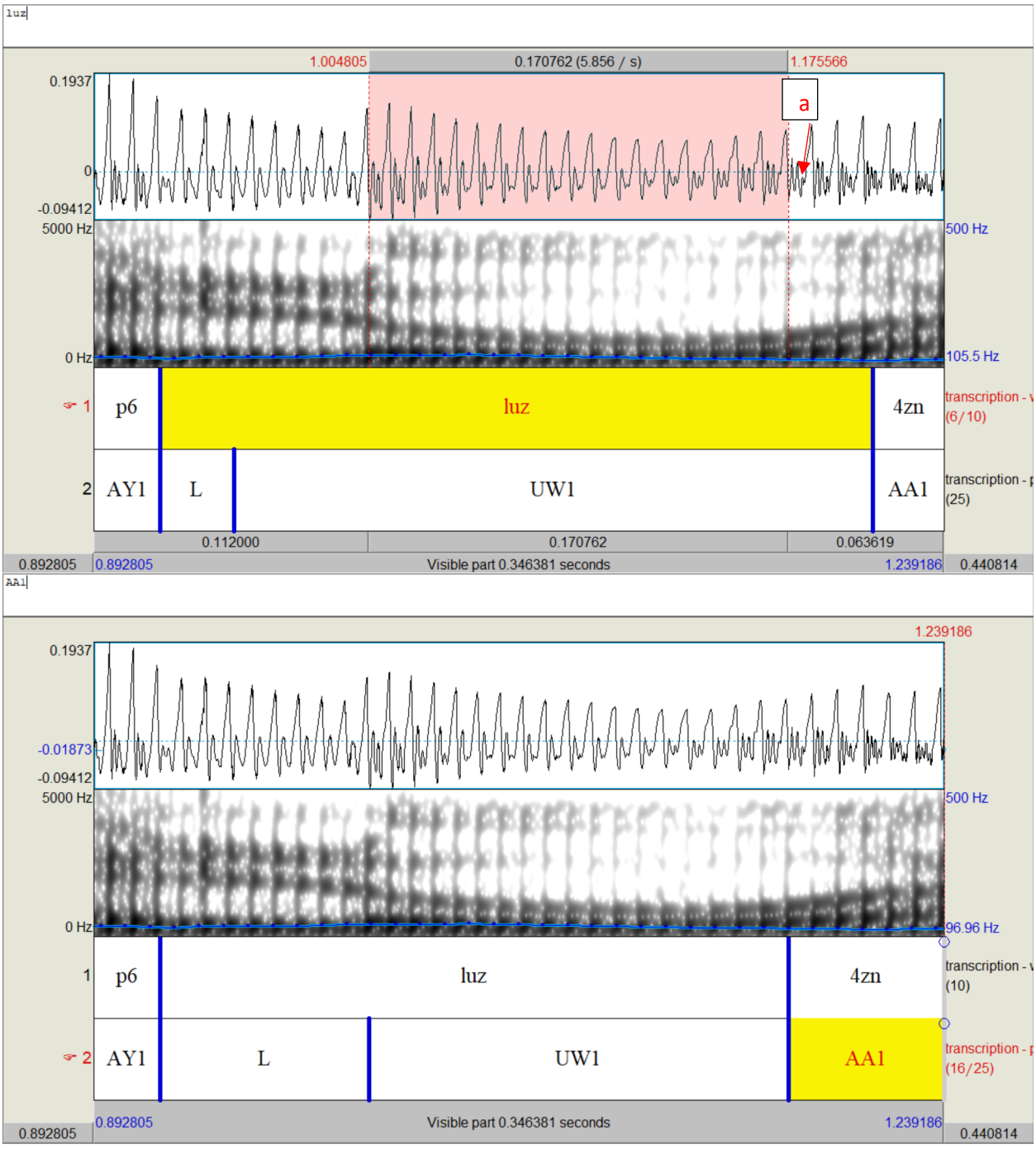


This segmented form is based off the bottom image above.



sp14_howl_1_6

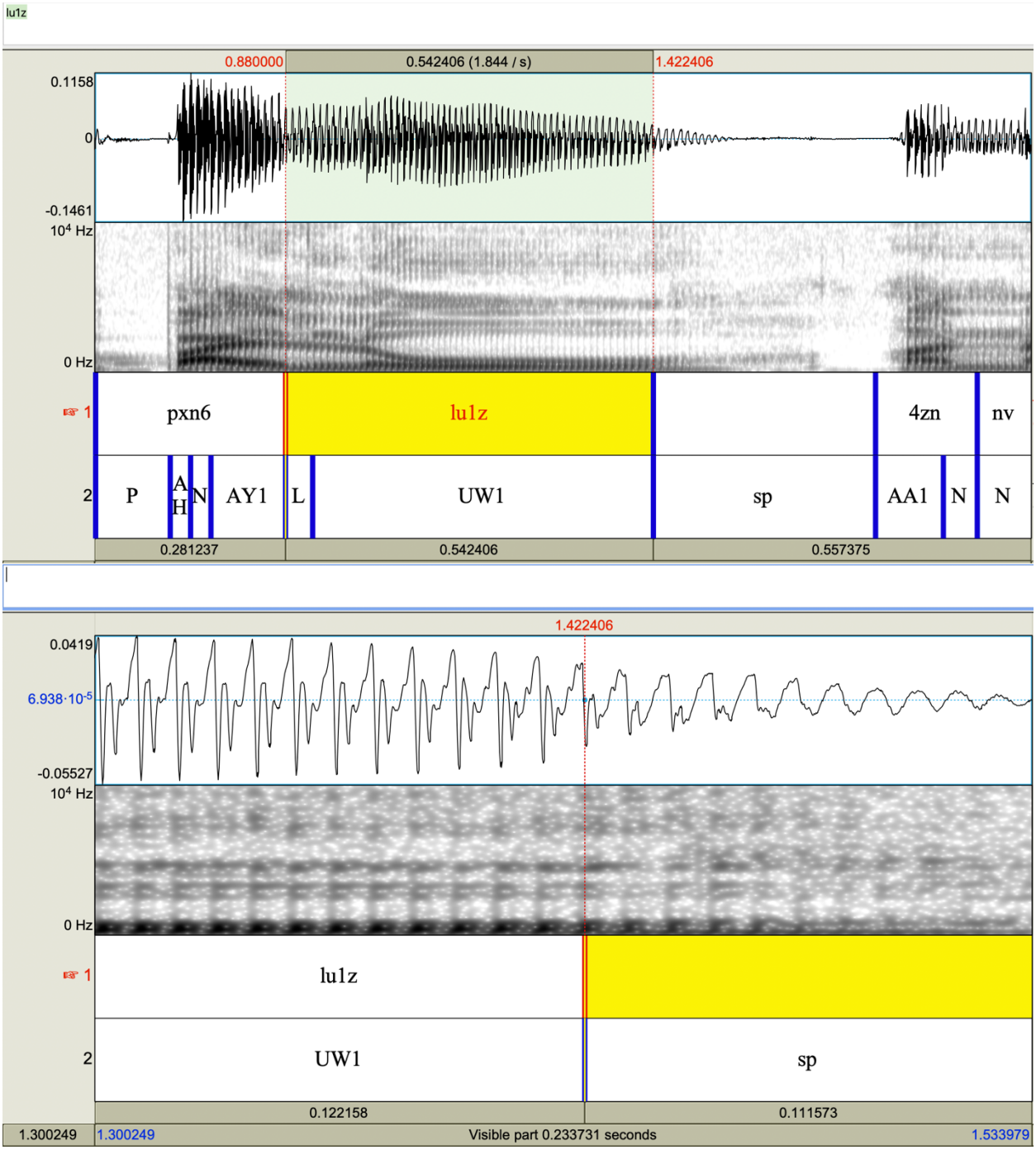
This one is quite tricky. Notice once again that [l] is way off. I choose the left end of the red selected portion as the beginning of the vowel because it is where F2 begins to steeply fall and also where the waveform begins to have 3 crests instead of 2. The endpoint is also tricky. The point I choose is because it is where F2 begins to curve up steeply and also where the wave begins to form a fourth crest between the original second and third crests (see where a is pointing). Notice that [l] has no VOT because it is a sonorant, not a stop!



sp35_thigh_1_9

Here we have an example where there is no coda and silence is what follows. The best place to put the boundary is at the point when the spectral energy really starts to die down and also when the waveform starts to change a lot. In the zoomed out view, you can see that the end of UW1 is when the 2nd and 3rd formants are really dying out, so it looks like a good point. In the second

image, zoomed in, you can see that the boundary works well because the waveform changes quite a bit (the crests change shape a lot).



(D) Analysis

(1) Let's organize some files. Create the following folders (replace "45" with your speaker number), assuming they don't already exist:

45-unaligned: Place the pre-MFA-force-aligned TextGrids
45-aligned: Place all the MFA-force-aligned TextGrids (i.e. those in the output folder that you created when you ran the forced alignment)
45-realigned: Place your newly realigned TextGrids in here.

Make sure you also have the following files:
45.eaf: the ELAN transcription
45.psfx: comes along with the ELAN transcription
45.wav: the .wav file of the speaker
45.TextGrid: the long Praat TextGrid you created from the ELAN file
45.xlsx: the metadata for the speaker
45_1to290.txt: the original transcriptions of each sentence, created when you ran the *save_intervals_to_wav_sound_files* Praat script
45-clips.txt: the clip names text file you created from the Excel file in order to name the files correctly
45-transcriptions.txt: the transcriptions from 45_1to290.txt transformed into MFA-friendly transcription by the transcriptions.ipynb jupyter notebook
demographic_info_thai_45: the PDF with the demographic data of the speaker
dictionary.txt: the dictionary for the MFA created from the transcriptions.ipynb jupyter notebook

***

Here is also a useful point to teach you some useful command line tricks.

If you want to replace all sequences in a filename with another one, check the instructions here for Windows: https://www.windowscentral.com/how-rename-multiple-files-bulk-windows-10#rename-files-using-command-prompt
For Mac, follow these instructions: https://unix.stackexchange.com/questions/19058/how-to-replace-one-char-with-another-in-all-filenames-of-the-current-directories

If you want to create a new folder named x in the current directory, type the following

```
mkdir x
```

If you want to move files, the command you want to type is mv [file] [folder]. We can use regular expressions (regex) to move all files of a certain type as well. This is a very useful trick to move all the .wav or .TextGrid files from one folder to another. In the line below, the regex character * is a wildcard meaning "0 or more of any character".

```
mv *.wav ../45-aligned
```

The above command will move all .wav files from the current folder to a folder called "45-aligned" that is contained in the folder that contains the current folder. Note that .. means "the containing folder". A similar command to **mv** is **cp**, which copies the given file instead of moving it. In Windows, **mv** is **move** and **cp** is **copy**.

(2) Make a folder in the same folder as where you put the transcriptions.ipynb file and name it **voicesauce**. Then put your folder with your realigned files in there. Make sure the name is in the right format ([speaker number]-realigned).

14-realigned

(3) Open up the notebook transcriptions.ipynb and run the first cell with your speaker number. Then run Part 3. When you run it, it will create files with a third interval tier (target), that will be populated with the target phone.

jupyter-lab transcriptions.ipynb

Let's do a sanity check on our output! When you run the cell that runs the function **create_target_tier** for each file, you will see each file, the gloss and target word (in MFA-friendly transcription), and the labels. Make sure of the following:

(a) The gloss matches the file name
(b) **Unless the word is 'to howl' [lu:] or 'thigh' [lu̟:], there should be 4 labels**
- The first and last should be blank ('')
- The second should be VOT (if it is 'to howl' or 'thigh', there is no VOT)
- The third should be the target vowel—make sure it is correct

(c) If there are more or less, something is off
- Maybe you added an extra label somewhere or are missing a label—fix that
- The speaker may have said the word more than once. If they did, please make a note in Sheet 2 of the speaker Excel file on the word and also write a note in the progress Excel file (especially if they hesitated on a word or something sounds off)

```
    # Write to TextGrid
    with open(f, 'w', encoding='utf-8') as f:
        f.write(tg.as_string('praat_long'))

    # Print extra line for readability
    print('\r')
```

```
sp6_athome_1_31
athome kuz
Label( t1=0.0000, t2=1.1274, text='b''' )
Label( t1=1.1274, t2=1.1731, text='b'VOT'' )
Label( t1=1.1731, t2=1.4215, text='b'uu'' )
Label( t1=1.4215, t2=2.3550, text='b''' )

sp6_athome_2_24
athome kuz
Label( t1=0.0000, t2=3.1903, text='b''' )
Label( t1=3.1903, t2=3.2201, text='b'VOT'' )
Label( t1=3.2201, t2=3.5927, text='b'uu'' )
Label( t1=3.5927, t2=4.6550, text='b''' )

sp6_athome_3_11
athome kuz
Label( t1=0.0000, t2=0.7856, text='b''' )
Label( t1=0.7856, t2=0.8161, text='b'VOT'' )
Label( t1=0.8161, t2=1.1505, text='b'uu'' )
Label( t1=1.1505, t2=2.2200, text='b''' )

sp6_athome_4_34
athome kuz
Label( t1=0.0000, t2=2.2627, text='b''' )
Label( t1=2.2627, t2=2.2972, text='b'VOT'' )
Label( t1=2.2972, t2=2.5664, text='b'uu'' )
Label( t1=2.5664, t2=3.5270, text='b''' )
```

Notice that howl below only has one label (‘uu’), since there is no VOT

```
sp6_howl_5_3
howl luz
Label( t1=0.0000, t2=2.3576, text='b''' )
Label( t1=2.3576, t2=2.6047, text='b'uu'' )
Label( t1=2.6047, t2=3.5080, text='b''' )
```

Below is an example of something that is off. Since there are two labels that say ‘a’, the speaker probably said the word twice. But since there is only one VOT, the annotator probably forgot to mark the second VOT. Go back and add it in and run the script again!

```
sp6_divorce_1_58
divorce tah
Label( t1=0.0000, t2=3.3634, text='b''' )
Label( t1=3.3634, t2=3.3873, text='b'VOT'' )
Label( t1=3.3873, t2=3.4560, text='b'a'' )
Label( t1=3.4560, t2=4.4000, text='b''' )
Label( t1=4.4000, t2=4.4500, text='b'a'' )
Label( t1=4.4500, t2=5.7550, text='b''' )
```

(3) Now you should move the .wav files into the “targets “ folder. You can use the command line trick you just learned to do so!

Mac: mv *.wav targets
Windows: move *.wav targets

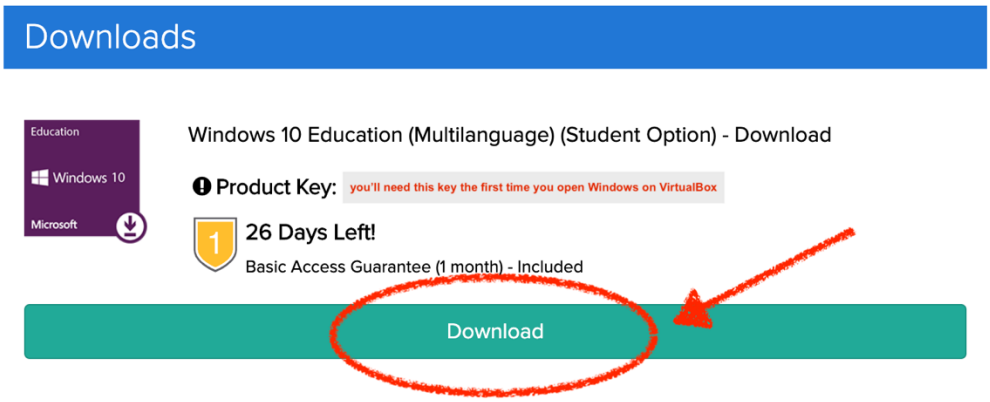
Now, if you do not have VoiceSauce please download it! If you have a Mac, you can set it up with the following guide written up by Nicole Kim!

Setting Up VoiceSauce

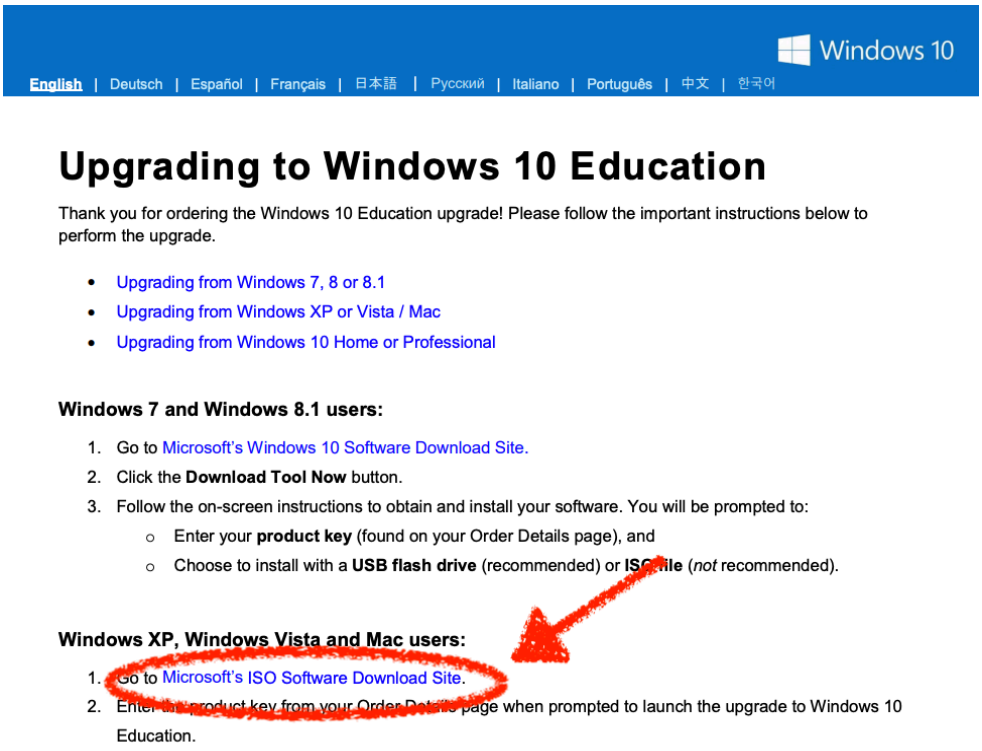
- Download Windows 10 & VoiceSauce
- Install Windows 10 in VirtualBox
- Install VoiceSauce

Step-by-step process below.

Download Windows 10



"Purchase" (it's free) Windows 10 (link).



Windows 10

English | Deutsch | Español | Français | 日本語 | Русский | Italiano | Português | 中文 | 한국어

Upgrading to Windows 10 Education

Thank you for ordering the Windows 10 Education upgrade! Please follow the important instructions below to perform the upgrade.

- Upgrading from Windows 7, 8 or 8.1
- Upgrading from Windows XP or Vista / Mac
- Upgrading from Windows 10 Home or Professional

Windows 7 and Windows 8.1 users:

1. Go to Microsoft's Windows 10 Software Download Site.
2. Click the **Download Tool Now** button.
3. Follow the on-screen instructions to obtain and install your software. You will be prompted to:
   - Enter your **product key** (found on your Order Details page), and
   - Choose to install with a **USB flash drive** (recommended) or **ISO file** (*not* recommended).

Windows XP, Windows Vista and Mac users:

1. Go to Microsoft's ISO Software Download Site.
2. Enter the product key from your Order Details page when prompted to launch the upgrade to Windows 10 Education.

The green *Download* button will open up a PDF.

Select "***Go to Microsoft's ISO Software Download Site.***"

Download VoiceSauce

Compiled Matlab executables - Windows 7/10
(v1.36 - Feb 27, 2019)

Matlab Component Runtime (32-bit)- MCR_R2015b_win32_installer.exe
Matlab Component Runtime (64-bit)- MCR_R2015b_win64_installer.exe

download this and this

VoiceSauce_bin.zip (6.4MB)

Instructions: Run MCRInstaller (only needs to be done once). Unzip VoiceSauce_bin.zip and run VoiceSauce.exe.

Note: Running VoiceSauce.exe for the first time may take a few minutes to load.

Link. Move files into a folder that you can later share with the new Windows 10 environment via VirtualBox.

Install Windows 10 in VirtualBox

(I followed this blog's instructions and it worked, here's the copy&paste:)

Open Virtual Box, click on "**New**" button.

Name your OS and select "Windows 10 (64-bit)" as OS version.

Select memory size. I accepted the recommended amount.

Create virtual hard disk.

Select hard disk file type. I chose the proposed VDI.

Select "Dynamically allocated".

Accept File Location and size.

It takes back to main screen. Select OS and click on Settings.

Go to "Storage" option.

Remove any empty optical drive that may exist. *(I don't think I had to do this step(?))*

Click on "Adds new storage attachment" > "Adds optical drive".

Select "Choose disk" in dialog.

Select ISO file download in previous step.

Run Windows 10

Select Windows 10 OS from VirtualBox main screen > ***Start***.

Enter Windows **product key** (provided when you first downloaded).

Windows installation screen will be displayed.

I chose the advanced option of just installing Windows.

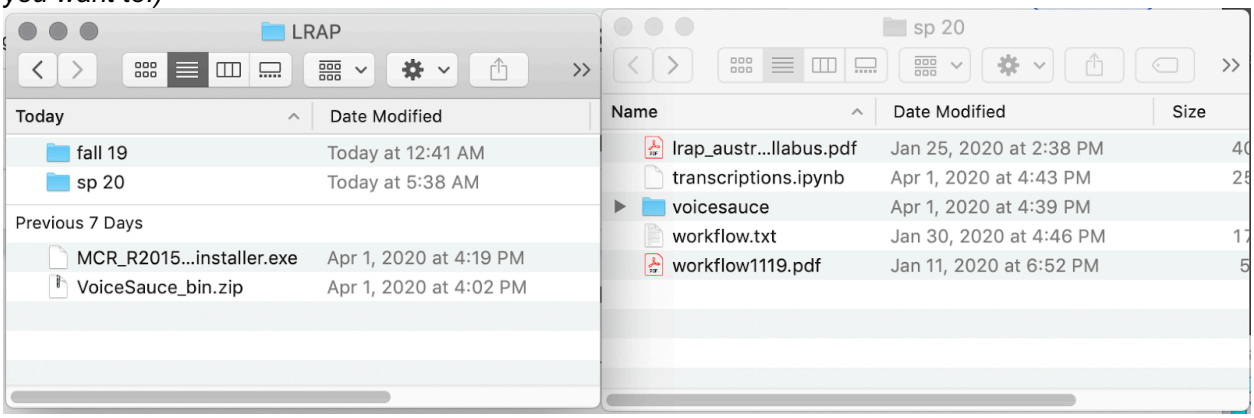
(I think somewhere here^, they asked for the .iso file… but I forgot exactly when.)

Installing VoiceSauce

Share folders - same process as BPM

On the VirtualBox start window, click ***Settings > Shared Folders***. Click ***Add New Folder*** icon (green plus sign).
Choose the folder containing MCR_R2015b_win64_installer.exe and VoiceSauce_bin.zip.

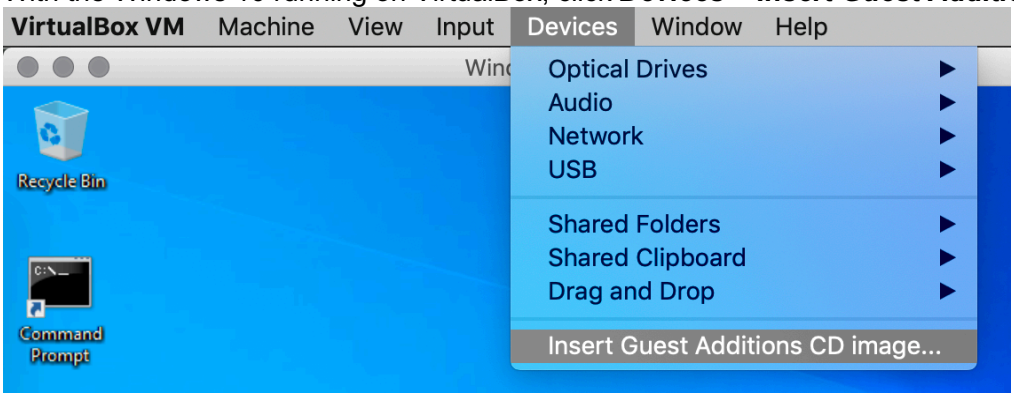
(I just decided to make an 'LRAP' folder containing these files + 'voicesauce' folder so that I won't need to repeat this step for the VoiceSauce process later on. You could [add another folder later] if you want to.)



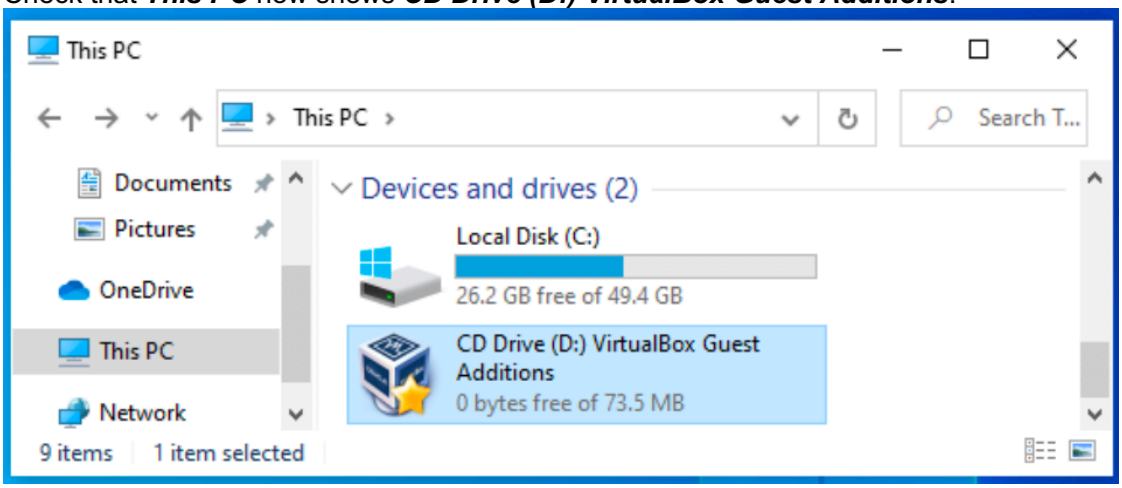
Accessing shared folders - install VirtualBox Guest Additions

(Took out bits of this and this page, with minor tweaks)

With the Windows 10 running on VirtualBox, click ***Devices*** > ***Insert Guest Additions CD image***.

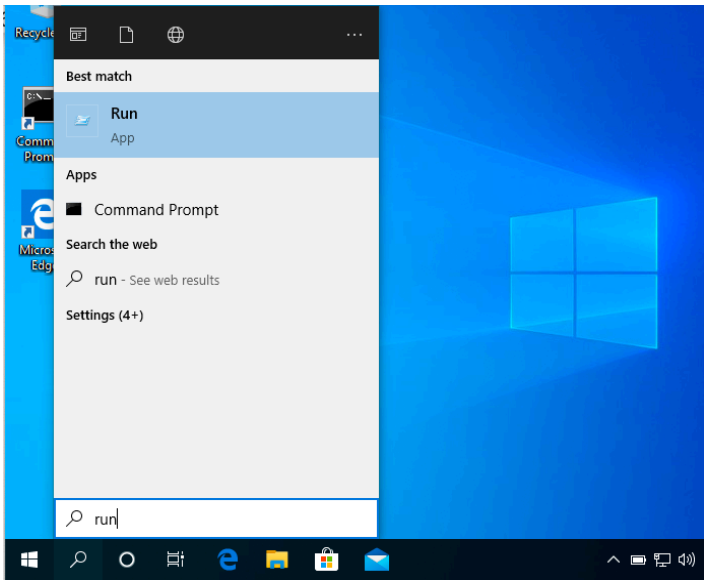


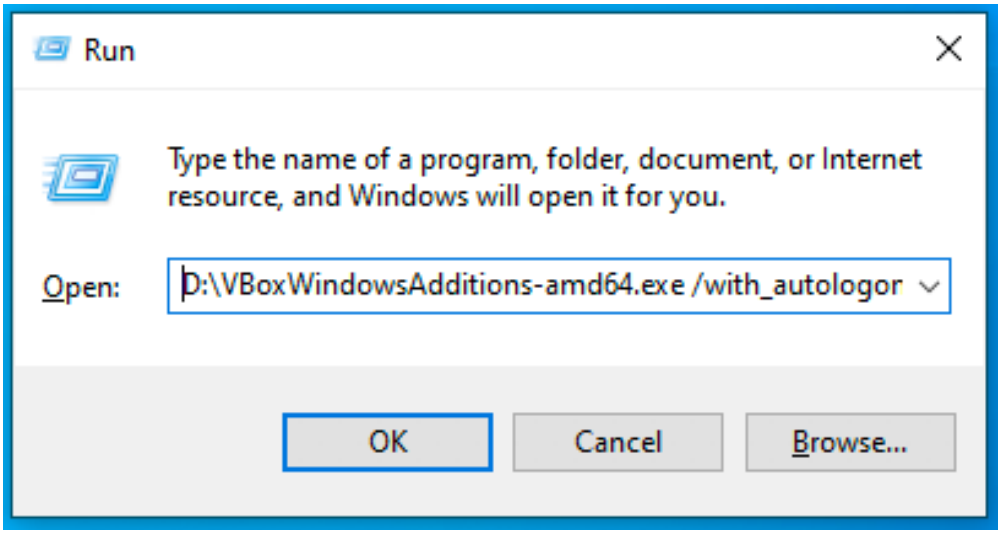
Check that ***This PC*** now shows ***CD Drive (D:) VirtualBox Guest Additions***.



Oracle: "You need to install the VirtualBox Guest Additions from the command line to get all the features we need." (I'm assuming this ensures that you don't have to repeat this process every time you restart the virtual machine. If you simply click on the installer to install, it won't include the autologon function)

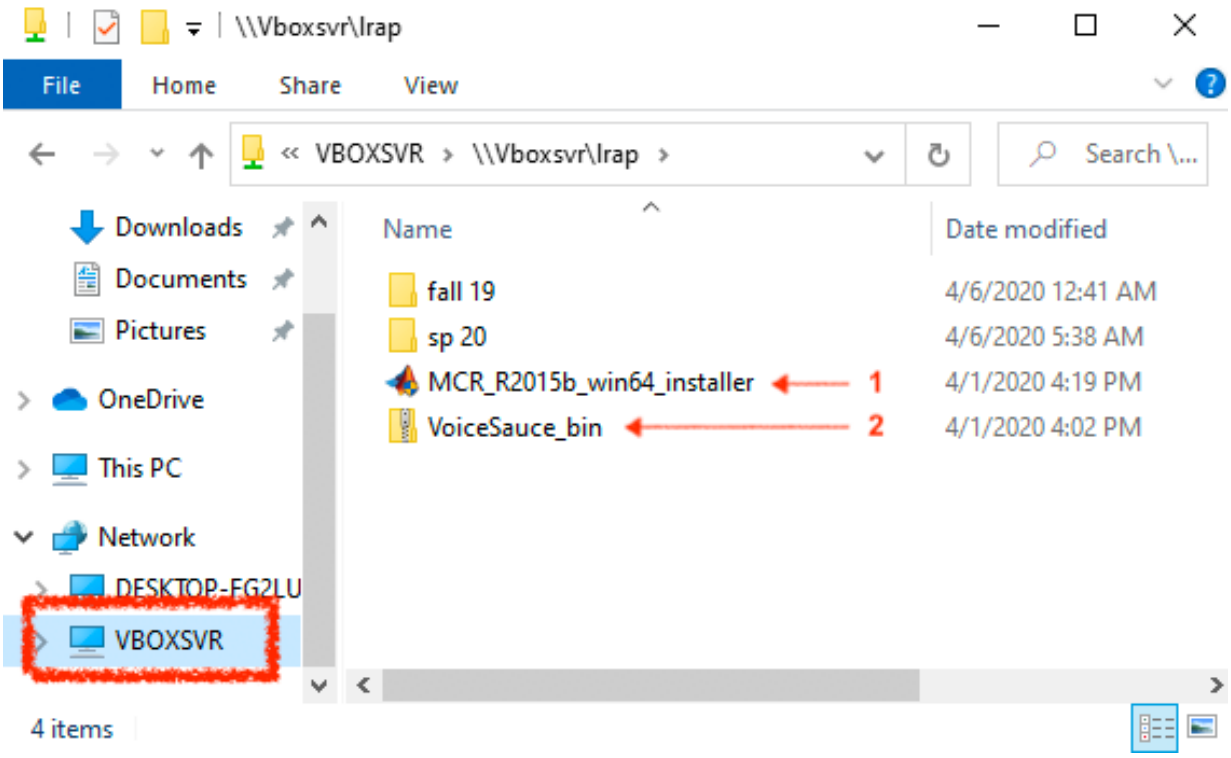
So, on the Windows Start menu, type **run** in the search field and press Return. The **Run dialog** is displayed. Enter **D:\VBoxWindowsAdditions-amd64.exe /with_autologon /with_vboxmmr** in the Open field.





When you are prompted, click **Yes** to install the Guest Additions. Click the **Next** button to move though the various steps of the VirtualBox Guest Additions Setup wizard. Accept all the default settings and then click **Install** to install the Guest Additions. Select **Reboot now** and click **Finish**.

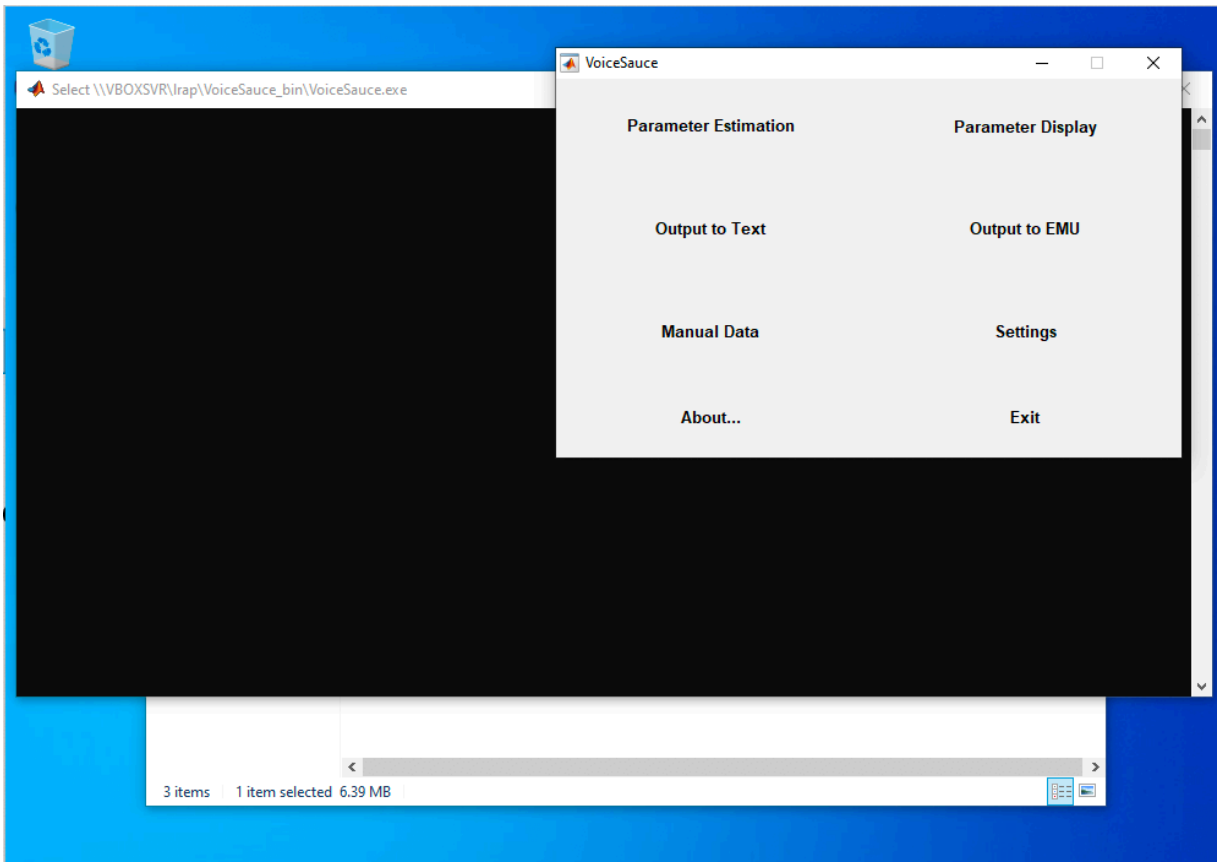
You should now see the shared folders appear as **network** file shares.



Set up VoiceSauce

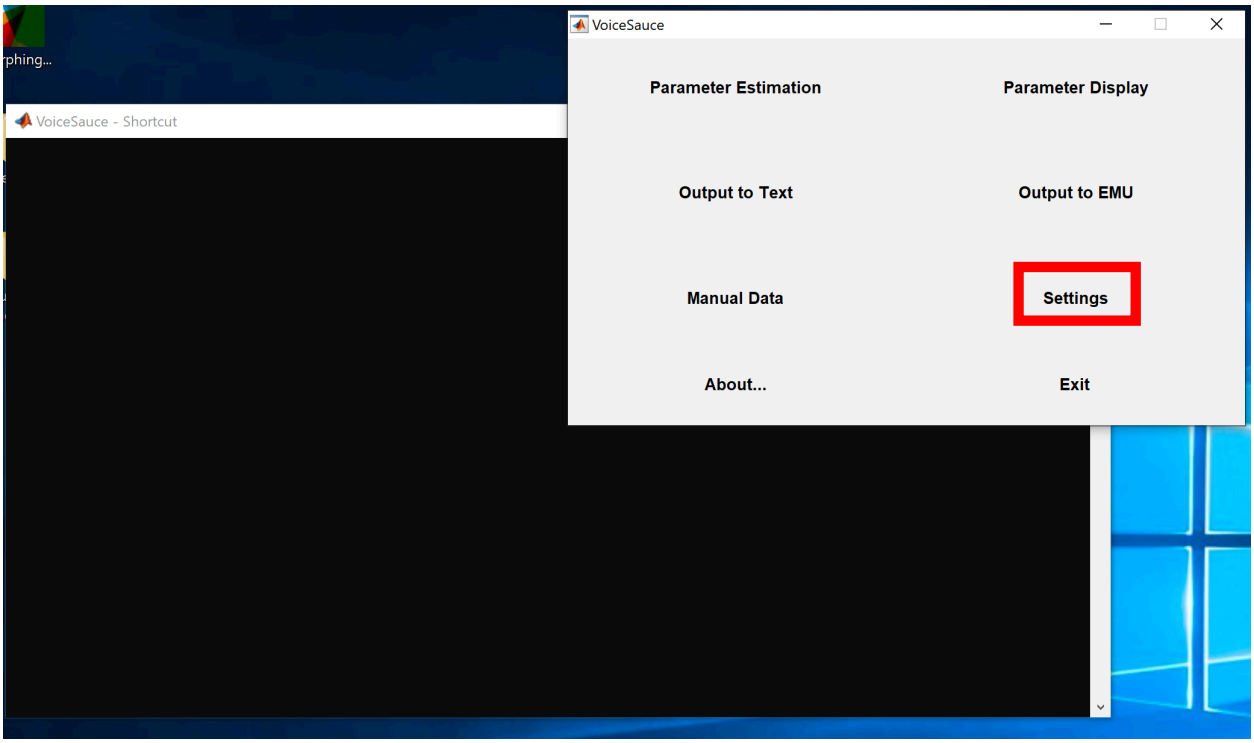
Install MCR first, then **unzip** the VoiceSauce_bin.zip file and you should be able to **open** VoiceSauce.exe.

It'll show just the black window for a while but eventually open the VoiceSauce start window.



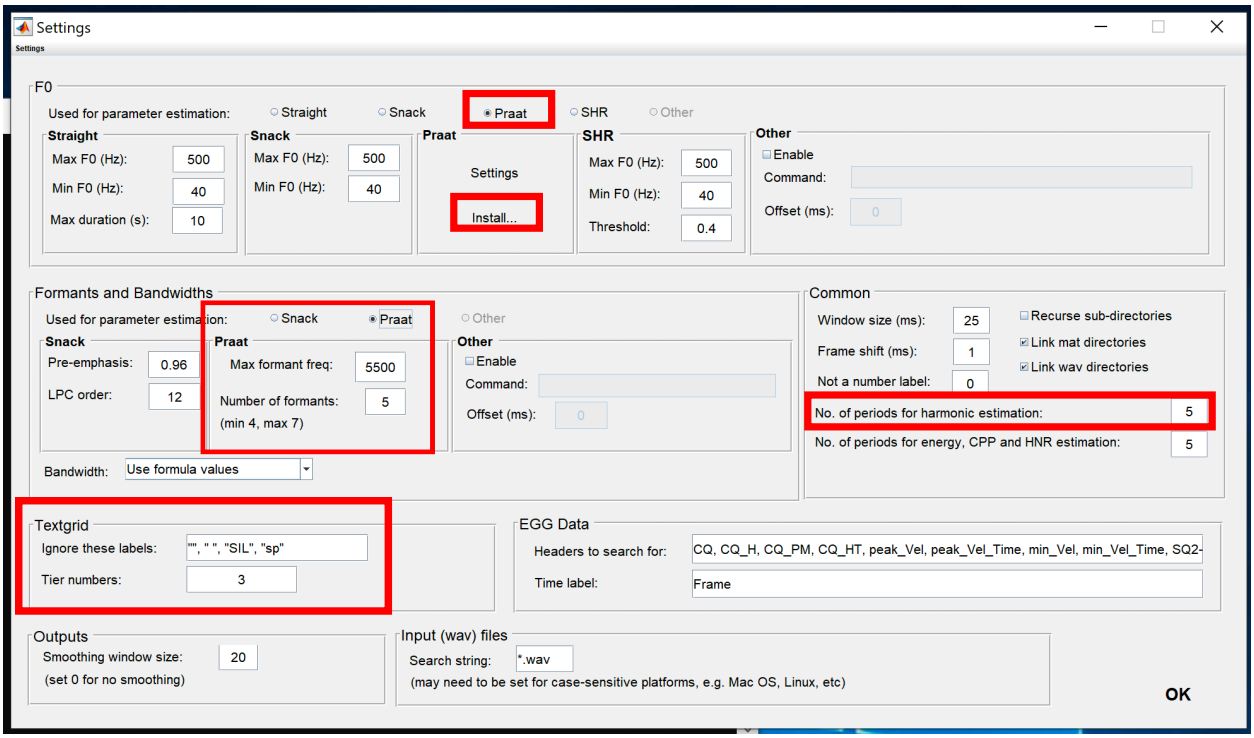
---

(4) Now open up VoiceSauce (in Windows VirtualBox if this isn't a Windows). You may see a black box for a while (up to a few minutes) but eventually you should see a number of options.

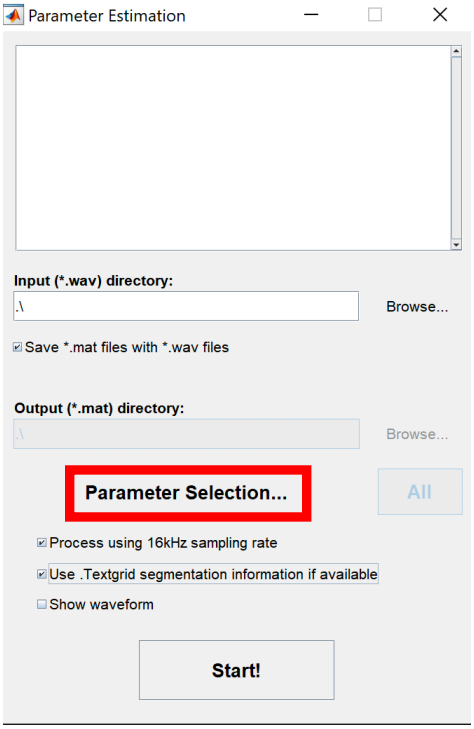


Go to Settings and make sure (you may have to click Install.. for Praat if this is the first time):

- The analyses are through Praat
- The tier analyzed is 3 (target)
- Ignore sp and sil (silences)
- Use 5 cycles for harmonic estimation
- Set it to Praat's default settings: 5500 Hz and 5 formants



(4) After pressing OK, go to Parameter Estimation in the main menu, choose the folder with the .wav and .TextGrid files as your input (*.wav) directory. Press the Parameter Selection button below.

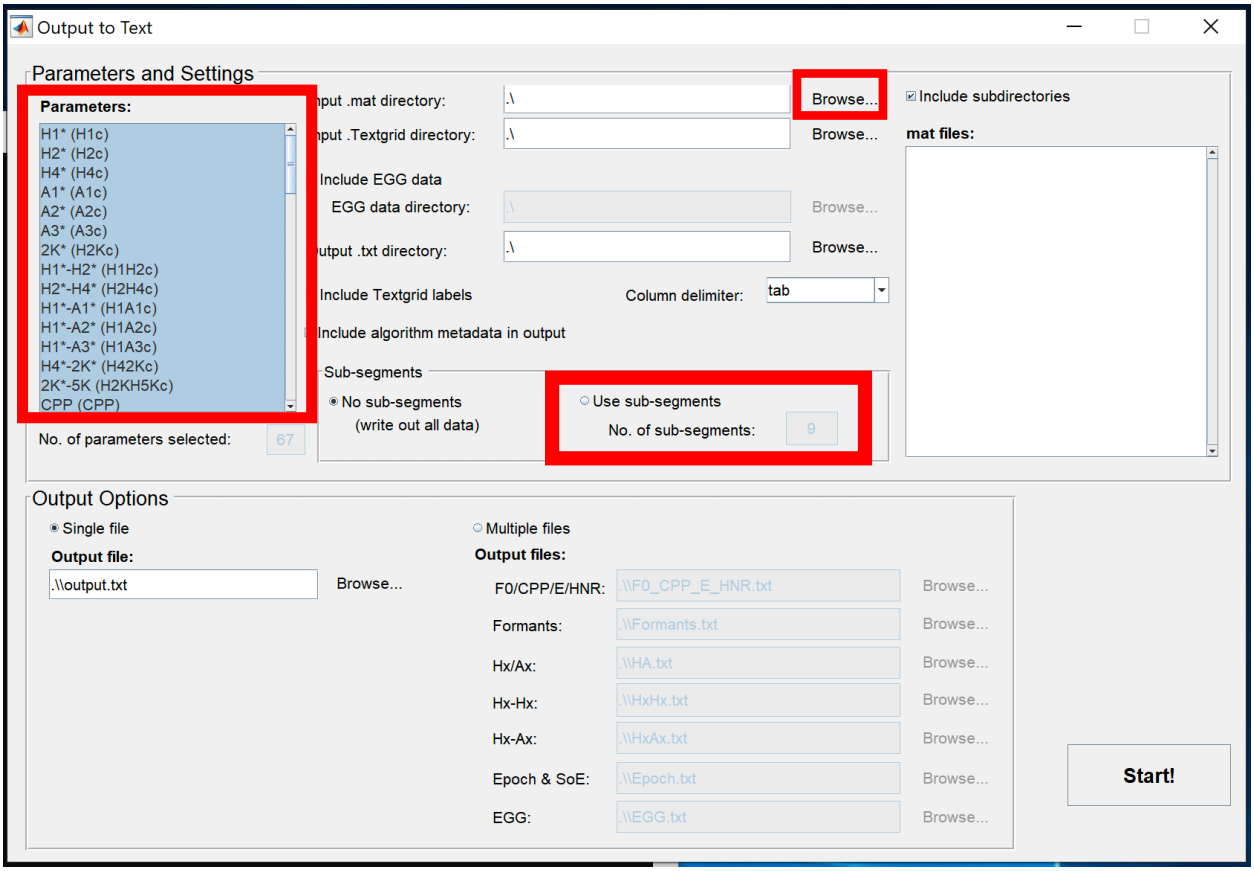


Deselect the Straight, Snack, and SHR estimations (you can deselect by just pressing those options once.

(5) Create a folder with your speaker number and the word ‘analyze’ (ex. 14-analyze) and move all the .mat files that have been created and the realigned .TextGrid files into it. Then move that folder into visualization/files folder in Box.

THE FOLLOWING IS ONLY IF YOU WANT TO LOOK AT YOUR INDIVIDUAL RESULTS—otherwise do not worry about this as we will look at everyone’s aggregated results together.

(6) After everything is estimated, go to Output to Text in the Main Menu and select the input .mat directory as the one that has all the .mat files you just created with Parameter Estimation. Once again deselect the non-Praat analyses. Then input how many subsegments to break it into (we will eventually want 4, 5, 10, and 20). And then press Start!



(7) Great, now you have the measurements and can analyze on pandas or R! You can play around with the visualize.ipynb script.