

Splicing

The object of splicing is to combine parts of two (or more) samples of speech to make them sound like a single sample. It is relatively easy to take two chunks of speech and concatenate them. The difficult part of splicing is making the concatenation sound natural.

In general, we choose to splice tokens in such a way (e.g., what we splice with what, and at what point in the signal) that we are most likely to be able to splice successfully. But there are still factors that can lead to difficulty or unnatural sounding splices:

- F0 (pitch) differences between tokens
- amplitude differences between tokens
 - o overall (one part is louder than the other)
 - o at the splicing point - misalignment of amplitude at a particular point in the waveform can lead to a spurious click in the spliced signal
- segment quality differences between tokens (although we generally try to splice between segments to minimize this difficulty)

Some basic principles:

- 1) Pick tokens that are as similar as possible with respect to pitch, quality, overall amplitude, duration, etc.
 - o Pitch (F0) is probably most important since amplitude and duration can be adjusted and segment quality won't play much role since you're splicing between segments.
- 2) Pick a splicing point at an amplitude 0-crossing. In other words, make sure that the waveform edges that you concatenate are both at 0.
 - o If the joined edges of the waveform don't meet up at 0, the signal may have a click.

Current project: /æ/ before velars

You will be trying to manipulate the vowel (/æ/) in monosyllabic words with velar stop codas (i.e., words of the type Cæg or Cæk). For the speaker in this study, /æ/ is raised before /g/ (so we will call it [ɛ] as a shortcut), but not before /k/.

- Your task is to create a set of Cæg words with a regular /æ/ in them so that the perception of words with this vowel pronunciation ([æ]) could be compared to the perception of words with the different but natural [ɛ].

Here's what we have:	t[æ]k 'tack'	t[ɛ]g 'tag'
Here's what we want:	t[æ]k	t[ɛ]g, t[æ]g

- In fact, however, you don't have to splice just the vowel from one word into another. You can take the whole CV from one word and splice it onto a new /g/ or /k/.

Splicing scheme:

	Cross-spliced	Same-spliced
Originals	$tæ_{\alpha}k$ $tæ_{\epsilon}g$	$tæ_{\epsilon}g$ $tæ_{\epsilon}g$
Spliced stimulus	$tæ_{\alpha}g$	$tæ_{\epsilon}g$

- Note that there are both cross-spliced and same-spliced tokens. Cross-spliced tokens are the newly invented token types. But in order to ensure that it is not just the fact of the splicing itself that leads to any perceptual differences we may find, we need to splice the tokens of the natural types too.
- But there's a remaining issue:
 - Vowels before voiced consonants (e.g., /g/) are naturally longer in duration than vowels before voiceless consonants (e.g., /k/). So if you take [tæ] from 'tack' and splice it onto [g] from 'tag', the resulting [tæɡ] will sound odd because the vowel will be too short. (Furthermore, since these items would likely be used in an experiment that measures response times to these items, we want the items (voiced and voiceless) to be of the same length.)
 - So we are going to neutralize the vowel duration difference by equalizing the vowel length in these sets to an intermediate value.

Splicing

- Choose a CVk and a CVg token from a minimal pair. Or for same-splicing, choose two different repetitions of a single CVk or CVg token.
- Select the relevant part in each file.
- In Praat, you have the following useful options:
 - move end/beginning of selection to 0, i.e., to a 0-crossing (in the Select menu in the Sound window)
 - copy selection (Ctrl-C)
 - paste *after selection* (Ctrl-V)
- You can develop your own method, but you want to do something like the following:
 - open both files
 - select the 2nd half of the composite item from the relevant token
 - move beginning of selection to 0
 - copy
 - select 1st half of the composite item from the other token
 - move end of selection to 0
 - extract selection (in the File menu)
 - in extracted selection (opened from object window), select all
 - I think there is no automatic way to select all—but the beginning doesn't matter so much, so if you select by hand by starting somewhere to the left and dragging off the window to the right, you should get the selection all the way to the end
 - paste (Ctrl-V) 2nd part (which should still be on the clipboard)

- Listen to the composite token to see if it sounds ok. If not, try again with a slightly different splicing point or with a different pair of original tokens, if you have them.

Duration manipulation

- Manipulate vowel durations for a whole set of tokens for a given minimal pair (CVg-cross-spliced, CVg-same-spliced, CVk-same-spliced) together.
- Measure the duration of the vowels in the voiced and voiceless tokens that you spliced together in the cross-spliced token.
- Split the difference between the voiced and voiceless vowel durations and equalize the durations of the tokens in the set to this intermediate duration.

Some basic principles:

- Only add or delete full cycles.
 - o This will most likely mean that you can't add or delete *exactly* the duration that you calculate. It will also mean that the durations of the 3 related tokens won't be *exactly* equalized, but they should be within about 6 ms (the approximate duration of 1 cycle) of each other.
- Spread the additions or deletions throughout the duration of the vowel (i.e., add or delete cycles at the beginning, middle, and end of the vowel rather than all in one place).
- When you add cycles, copy a cycle adjacent to the point where you will insert it.
- Make sure to match 0-crossings when you add or delete cycles.
 - o This is pretty easy if you use the move beg/end of selection to 0-crossing feature in the Select menu. When you are adding a cycle, you can simply select the cycle, copy (ctrl-C), and immediately paste (ctrl-V) to put the new cycle at the end of the last cycle (since you will already have selected up to the 0-crossing and the paste function inserts after the selection).

Illustrations with attached sound files

- creating a cross-spliced *bag* from *back* and *bag*

bæ_æk & bæ_æg → **bæ_æg**

- o Click on the speaker icons in the electronic version of this document to hear the soundfiles.

Figure 1 *back* - This will be the 1st half of the composite spliced item. (Notice that I didn't include the little breathy/devoiced portion at the end of the vowel in the part of the vowel to be spliced. This kind of devoicing is generally only found before voiceless stops, so it would sound unnatural to splice it into a voiced stop context.)

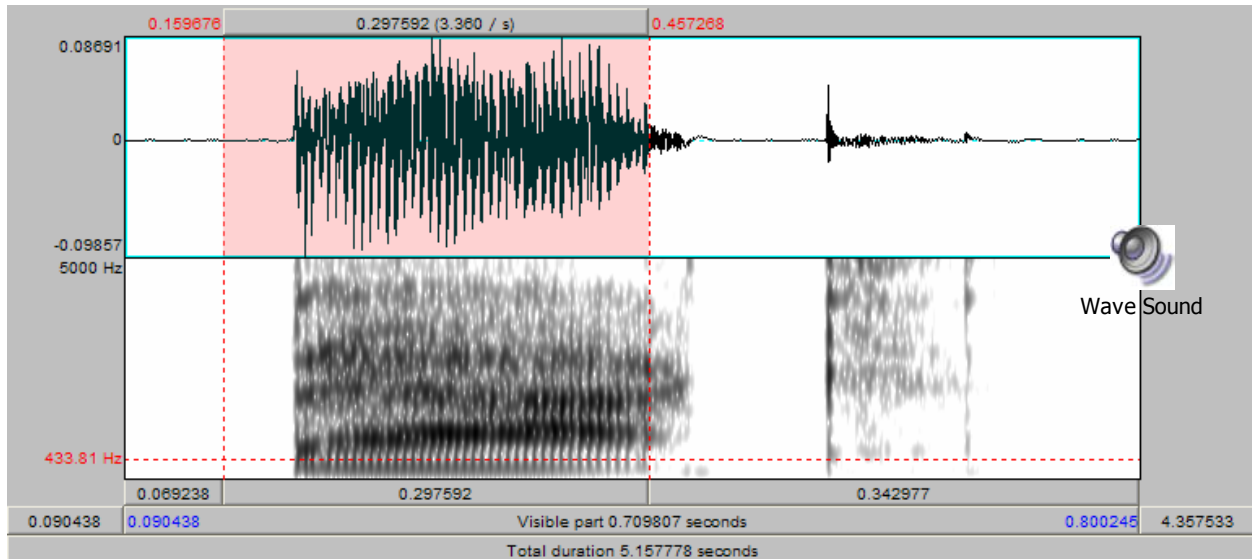


Figure 2 *bag* - This will be the 2nd half of the composite spliced item. (NB. For a voiced stop, this /g/ has a quite strong release burst. After hearing the resulting composite token, I think this may not have been the best choice as this type of release is more typical of a voiceless stop.)

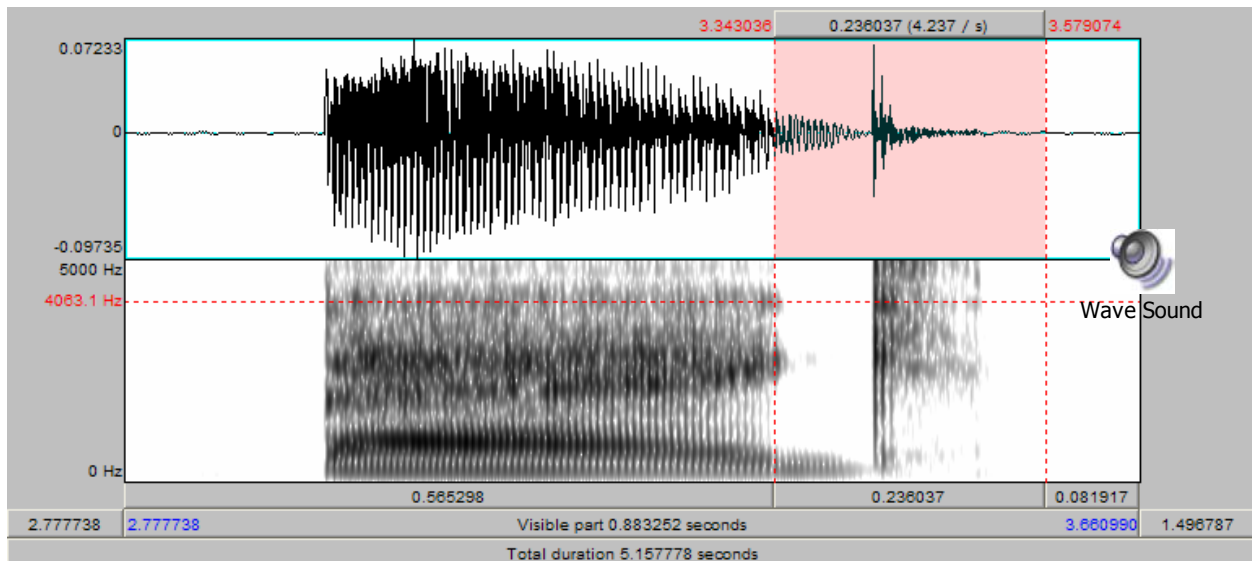


Figure 3 Initial spliced token - This is the result of pasting the /g/ of *bag* to the /bæ/ of *back*. The cursor marks the splice point.

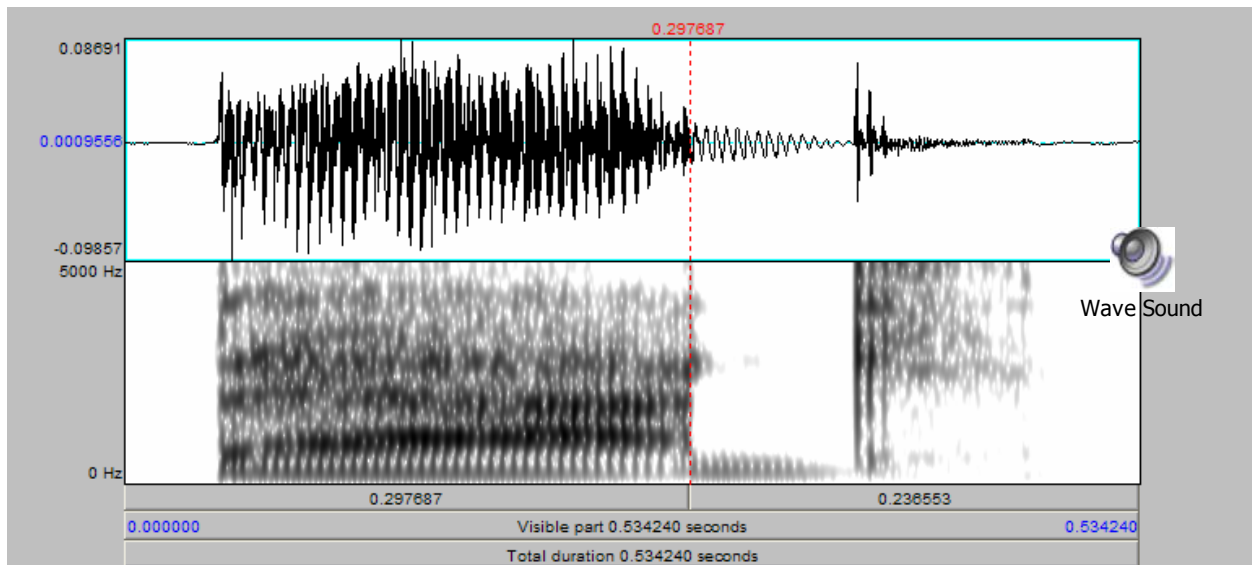


Figure 4 Close-up of the splice point - (Note that the splice point is at a 0-crossing, so there is no sudden jump or discontinuity in the waveform.)

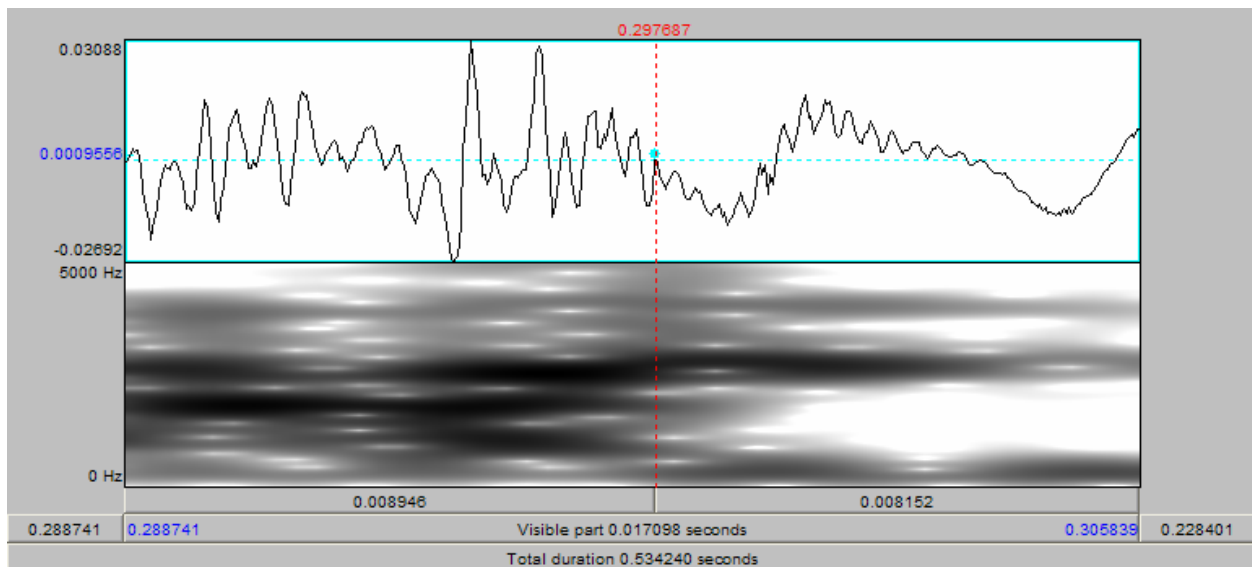


Figure 5 Vowel durations - These selections show the vowel durations for *back* (in the top figure) and *bag* (in the bottom figure). /æ/ in *back* is 256 ms. /æ/ in *bag* is 397 ms. The difference is 141 ms, so to split the difference, we will want to add 70 ms to the /æ/ _{back} here and in the same-spliced *back* token (and where /æ/ _{bag} is used, e.g., in the same-spliced *bag* token, we will want to delete 70 ms).

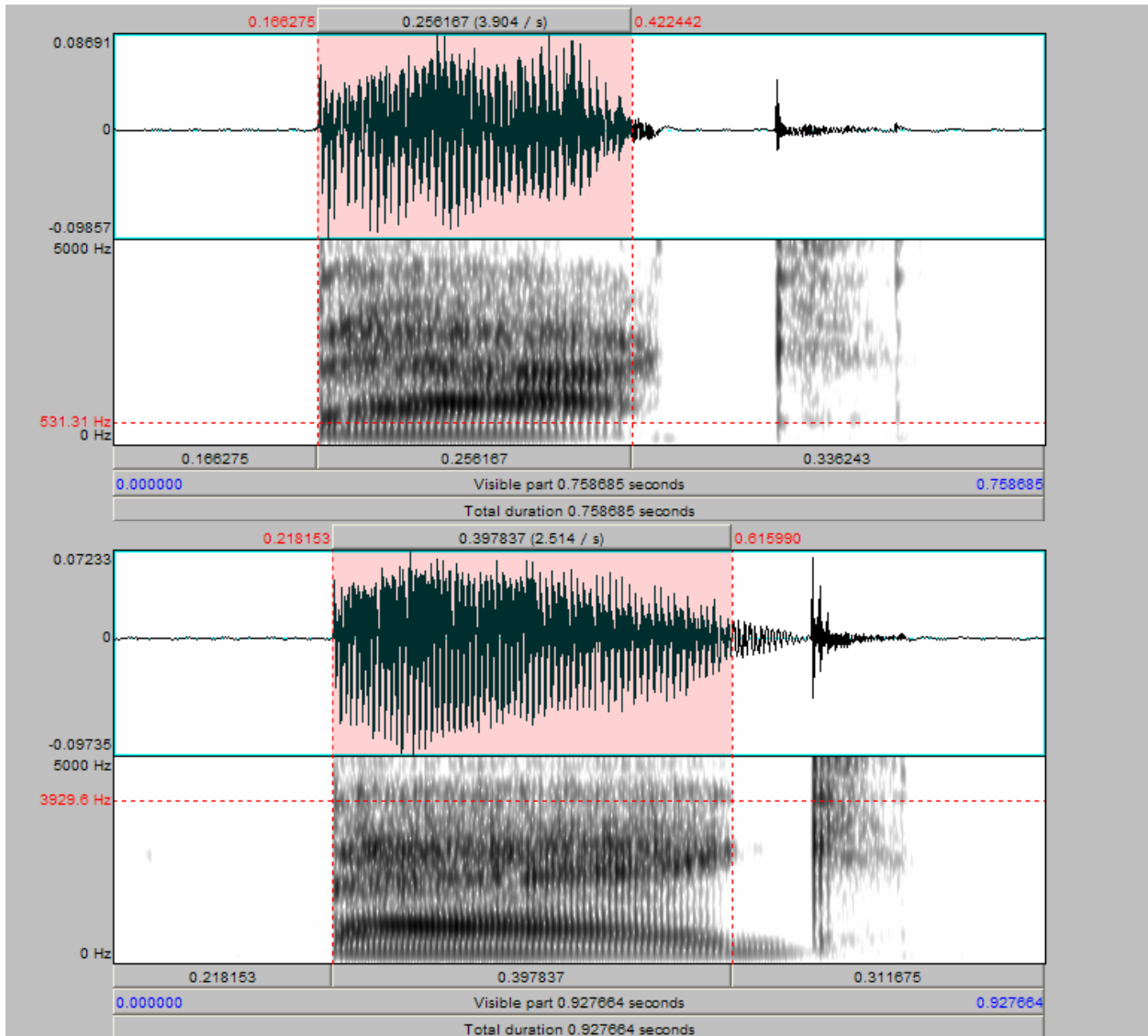


Figure 6 Final spliced token - This is the resulting spliced token with a manipulated intermediate vowel duration. (To me, this still sounds a bit like *back*. This may be partly due to the stop release, as I mentioned with Figure 2. It may also, of course, be due to the vowel quality, which is what we want to find out.)

