

PraatSauce/VoiceSauce comparison: White Hmong (male speaker)

James Kirby

31/01/2018

Contents

Data	1
Plots	2
F0	2
Formants	3
Bandwidths	5
Uncorrected amplitudes	9
Corrected amplitudes	11
Corrected differences	21
Cepstral peak prominence	24
Harmonic to noise ratios	25
Distinguishing voice qualities	26
High vowels	26
Low vowels	29

Data

This small dataset compares spectral measures generated by both PraatSauce v0.2.2 and VoiceSauce v1.31 at 1 msec intervals for 9 White Hmong lexical items spoken by a single male speaker. The original audio files can be found [here](#). For both scripts, 5 formants were estimated with a maximum formant frequency of 5000 Hz; minimum and maximum F0 values were set to 50 Hz and 300 Hz for all F0 estimators. For VoiceSauce, the STRAIGHT F0 estimate and Snack formant/bandwidth estimates were used for harmonic amplitude corrections.

The **method** column indicates whether the formant bandwidths were **estimated** using Praat (PraatSauce) or Snack (VoiceSauce), or whether the Hawks and Miller **formula** was used.

Note that in Hmong orthography, final *-g* indicates a low-falling breathy tone, while *-m* indicates a creaky tone.

```
head(df)
```

```
##      Filename Item Label seg_Start seg_End   t_ms      t method
## 1 12-cab-w_Audio cab    a   648.166 902.471 648.166 0.000000000 formula
## 2 12-cab-w_Audio cab    a   648.166 902.471 649.166 0.003952569 formula
## 3 12-cab-w_Audio cab    a   648.166 902.471 650.166 0.007905138 formula
## 4 12-cab-w_Audio cab    a   648.166 902.471 651.166 0.011857708 formula
## 5 12-cab-w_Audio cab    a   648.166 902.471 652.166 0.015810277 formula
## 6 12-cab-w_Audio cab    a   648.166 902.471 653.166 0.019762846 formula
##      script measure value corrected
## 1 PraatSauce    pF0 139.762 uncorrected
## 2 PraatSauce    pF0 139.870 uncorrected
## 3 PraatSauce    pF0 139.979 uncorrected
## 4 PraatSauce    pF0 140.088 uncorrected
```

```
## 5 PraatSauce      pF0 140.197 uncorrected
## 6 PraatSauce      pF0 140.306 uncorrected
```

In the plots which follow, the PraatSauce measures are unsmoothed. If you want to compare to smoothed estimates, uncomment the two lines:

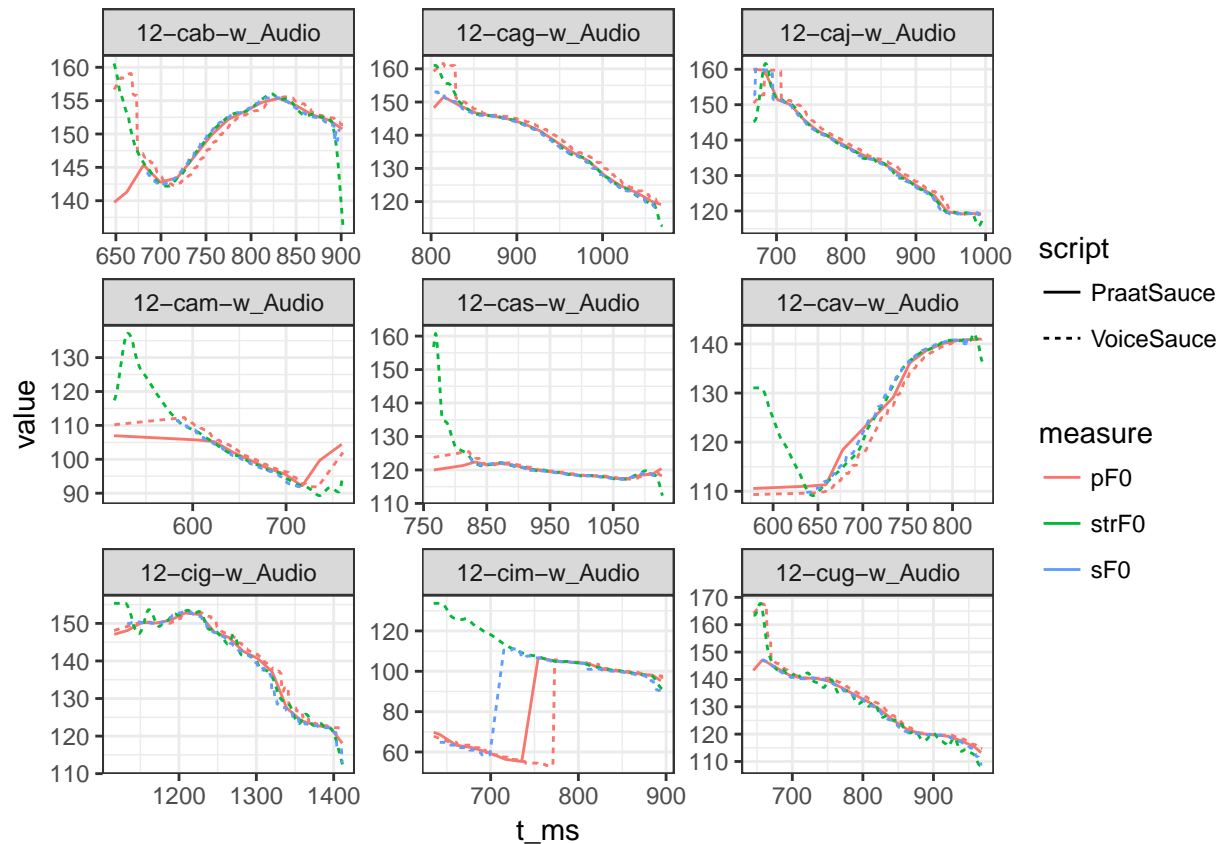
```
ps.fbw <- cbind(ps.fbw[1:6], apply(ps.fbw[7:43], 2, filter, filter=f21, sides=2))
ps.ebw <- cbind(ps.ebw[1:6], apply(ps.ebw[7:43], 2, filter, filter=f21, sides=2))
```

This implements a symmetric kernel filter. This is different from what VoiceSauce does. VoiceSauce uses the Matlab `filter()` function, by default a lag filter which pads with zeros. So while the smoothed value of sample 20 is equal to $\sum_{i=1}^{20} / 20$, the smoothed value of sample 19 is not undefined, but is calculated as $\sum_{i=1}^{19} / 20$, the smoothed value of sample 18 is $\sum_{i=1}^{18} / 20$, etc.

If you want to smooth the Matlab way, use the lag kernel by selecting `filter=f20` and set `sides=1`.

Plots

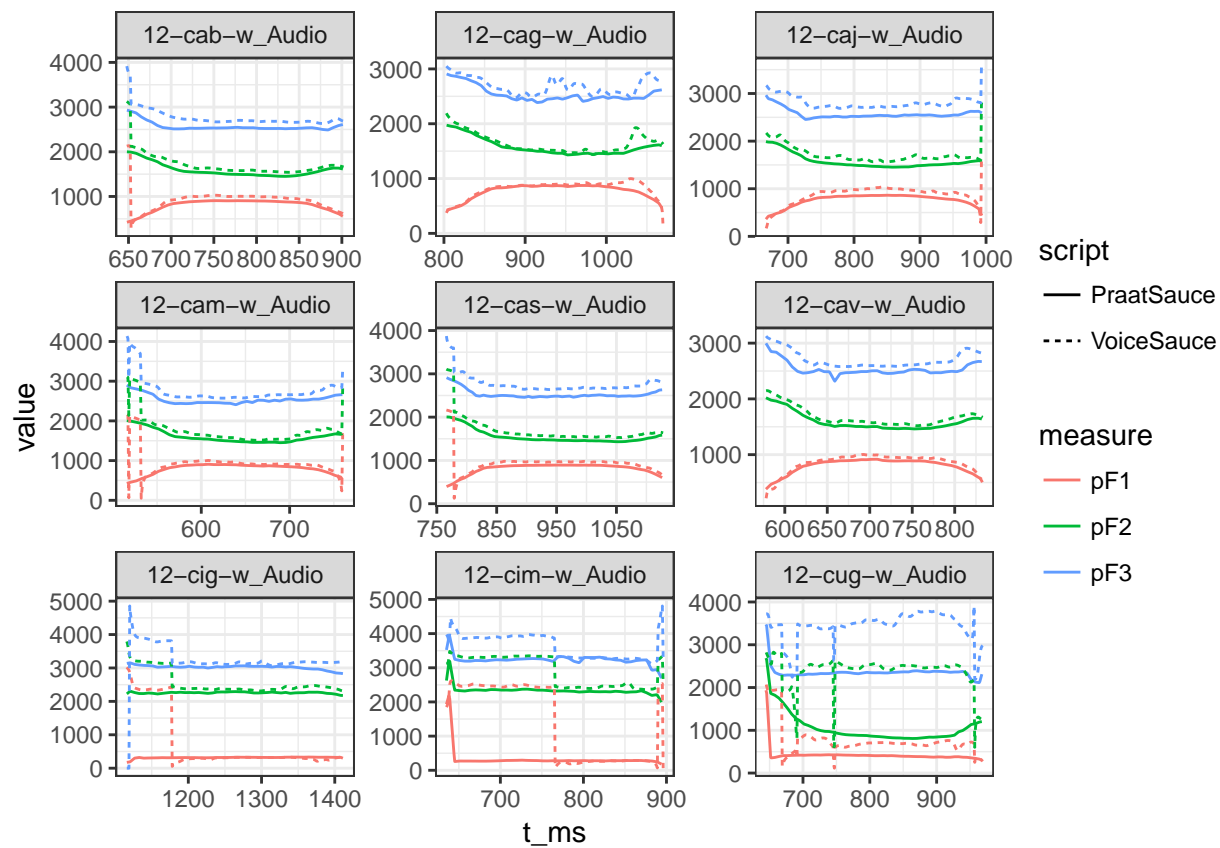
F0



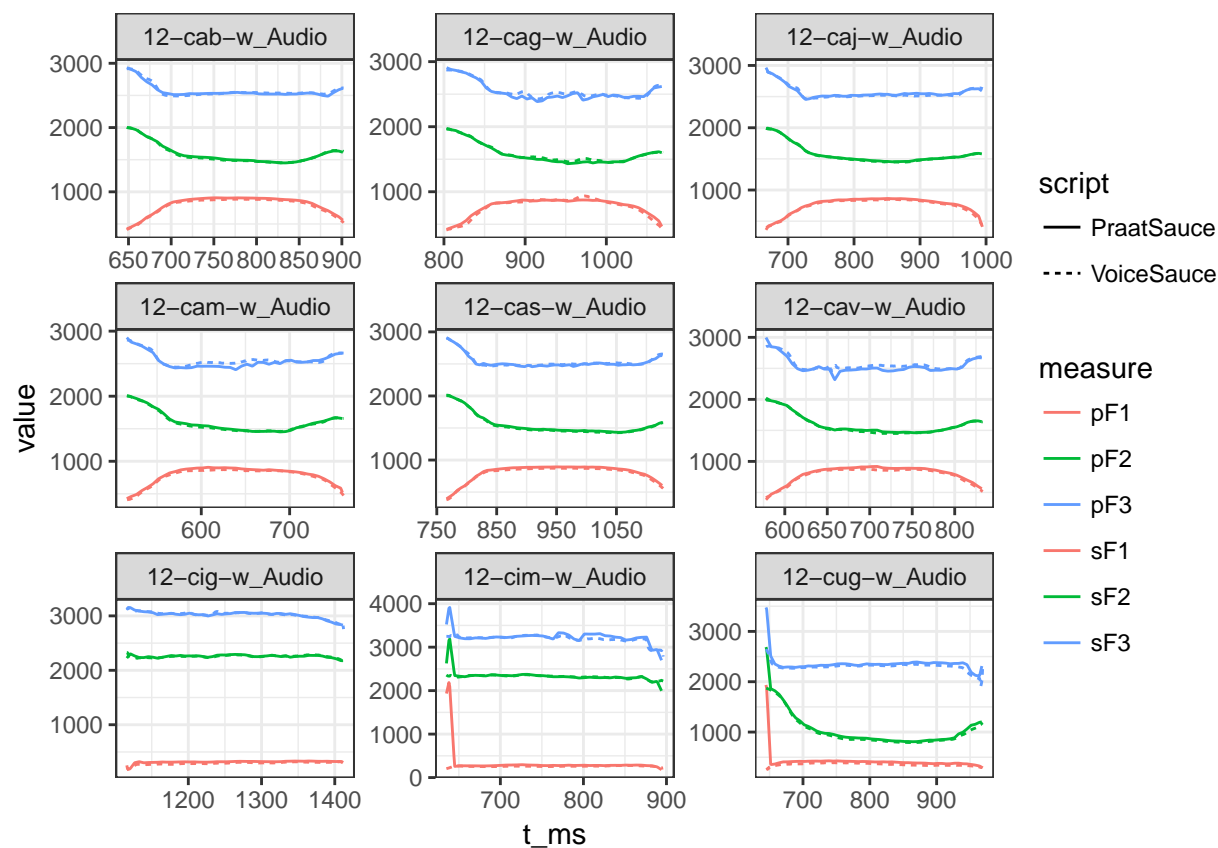
STRAIGHT appears to be capturing CF0 effects that most other estimators or not. This is an example of where pitch settings can be important: if the default PraatSauce pitch settings are used (40 Hz and 600 Hz), PraatSauce consistently fails to detect the initial F0 perturbations.

Formants

Formants estimate using Praat



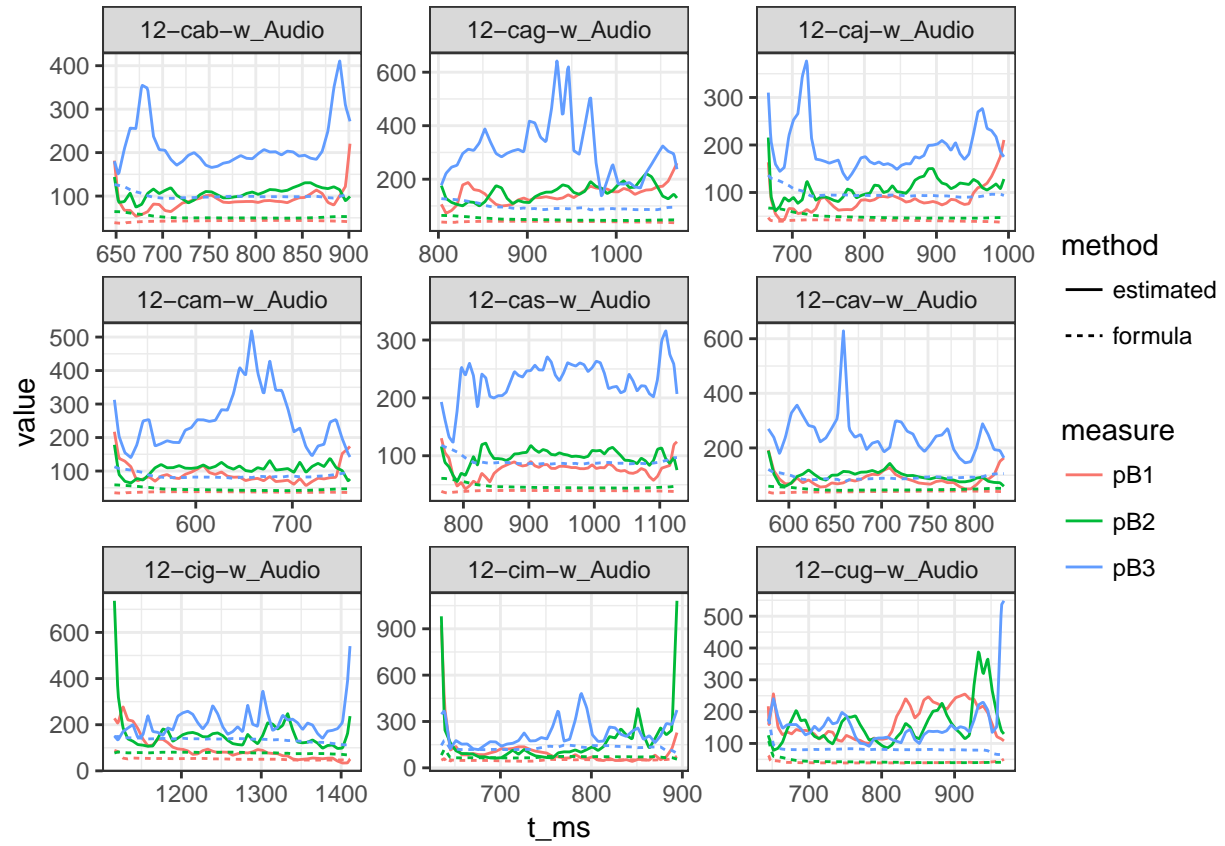
Formants estimated using Snack (VoiceSauce) vs. Praat (PraatSauce)



There are a few difference, especially at edges, some of which may be due to smoothing. However, it's not clear why the Praat-based estimates aren't identical: both scripts use the exact same command, with the same parameters, to estimate the formants (and bandwidths).

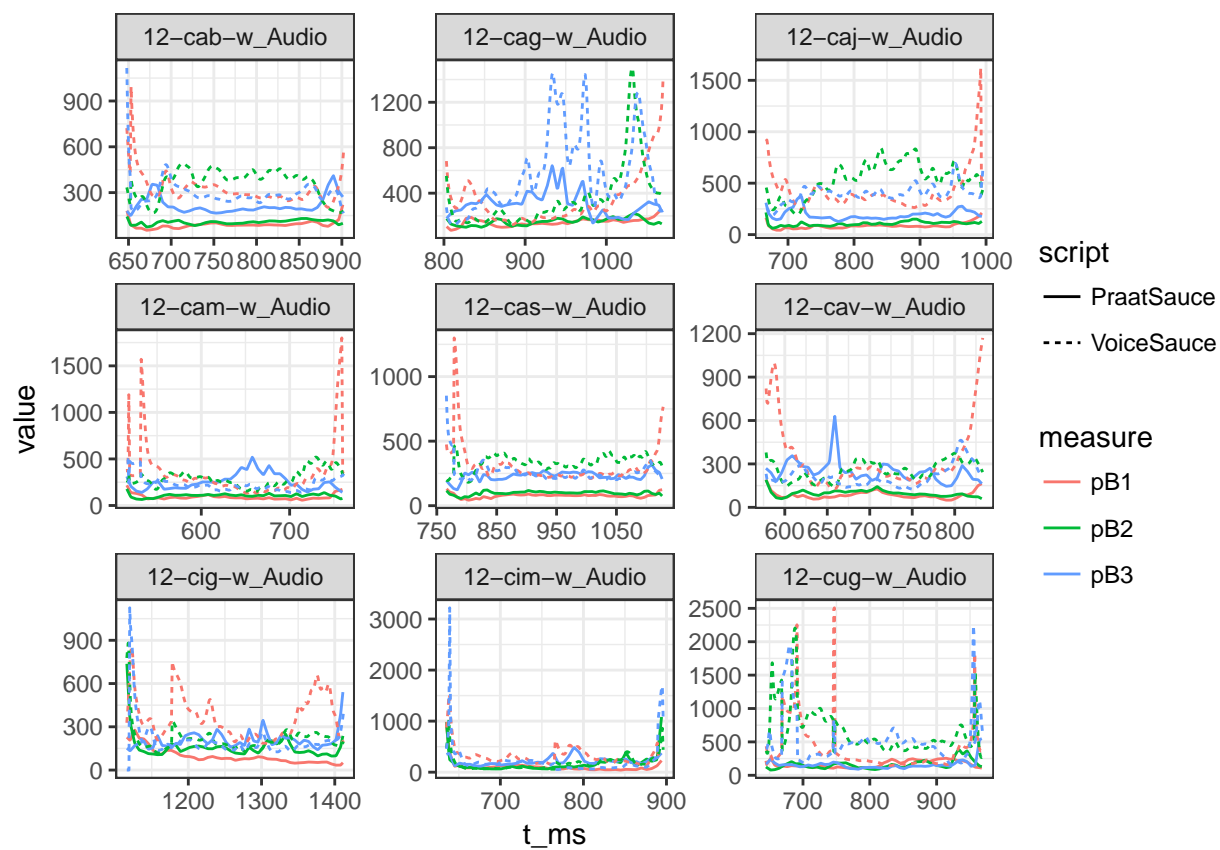
Bandwidths

PraatSauce estimated vs. formula bandwidths



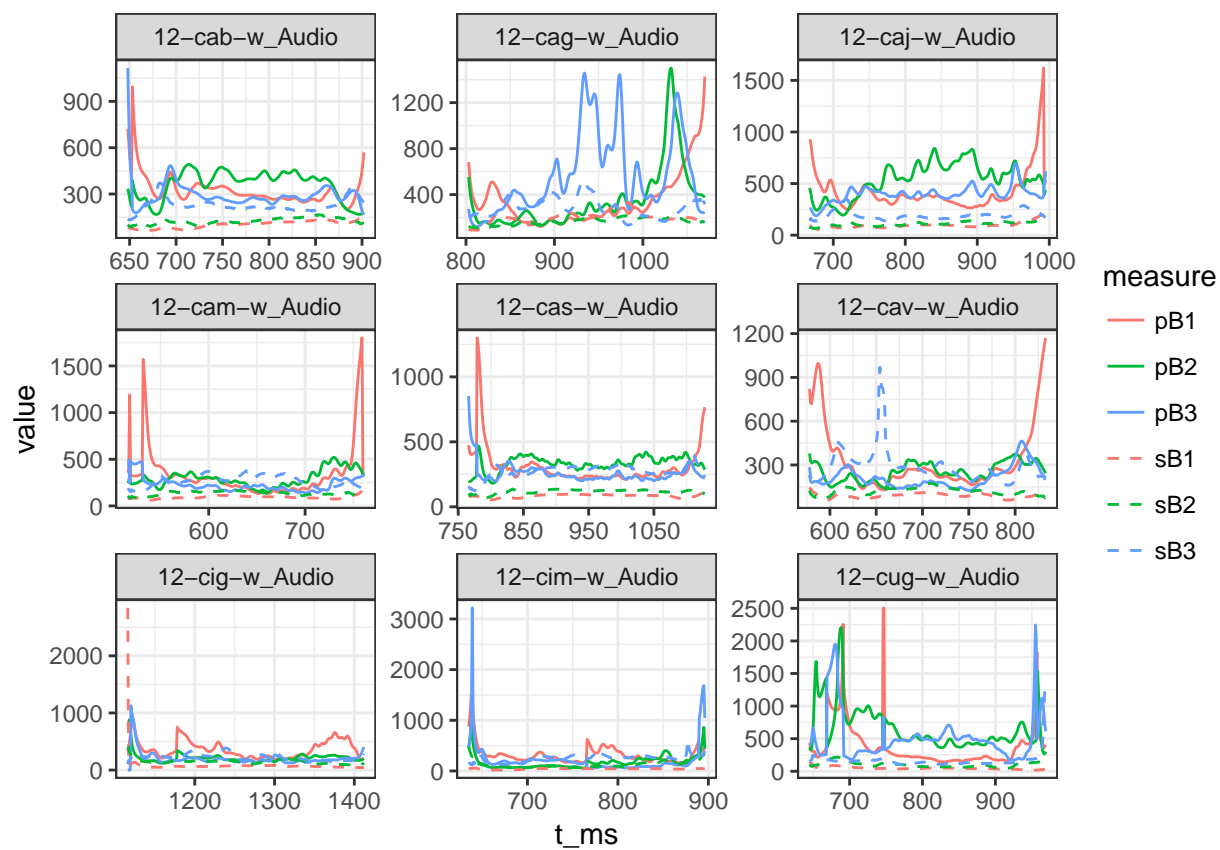
PraatSauce estimated bandwidths are huge...

PraatSauce vs. VoiceSauce estimated bandwidths



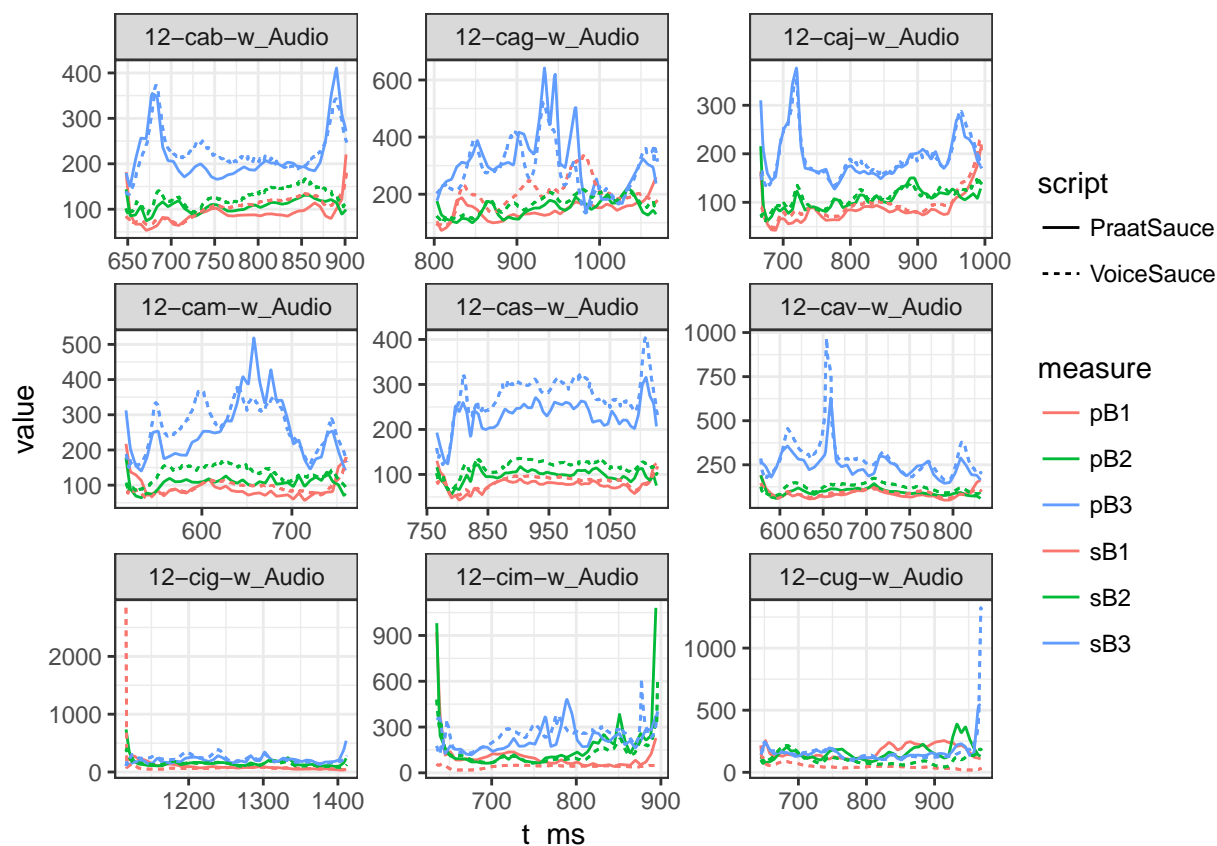
... but VoiceSauce Praat-estimated bandwidths are frequently an order of magnitude huger.

VoiceSauce Praat vs. Snack estimated bandwidths



VoiceSauce's Snack estimates (if that's really what they are) look less erratic.

VoiceSauce Snack vs. PraatSauce estimated bandwidths



Once again, the degree of overlap between PS-Praat and VS-Snack makes me wonder if the VS estimates aren't getting reversed somehow in the output, though I can't find any obvious evidence that this is the case in the VS code. However, it does appear that the way VS "uses" Praat formant estimates to estimate bandwidths is by taking the formant estimate and applying the Mannell (1998) formula

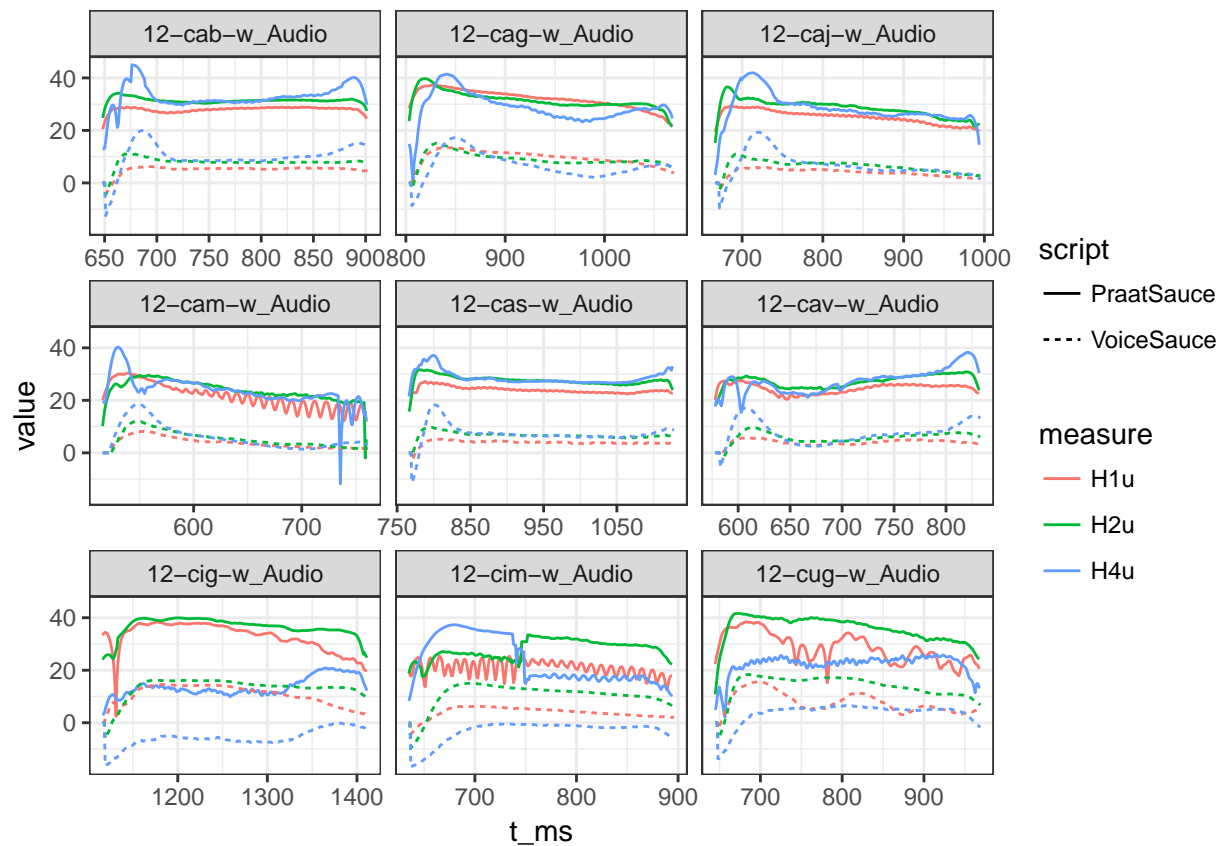
$$b_i = 80 + 120f_i/5000$$

while PraatSauce uses Praat's estimates of formant bandwidths, which appear to be a fixed function based on the frequencies of the adjacent formants.

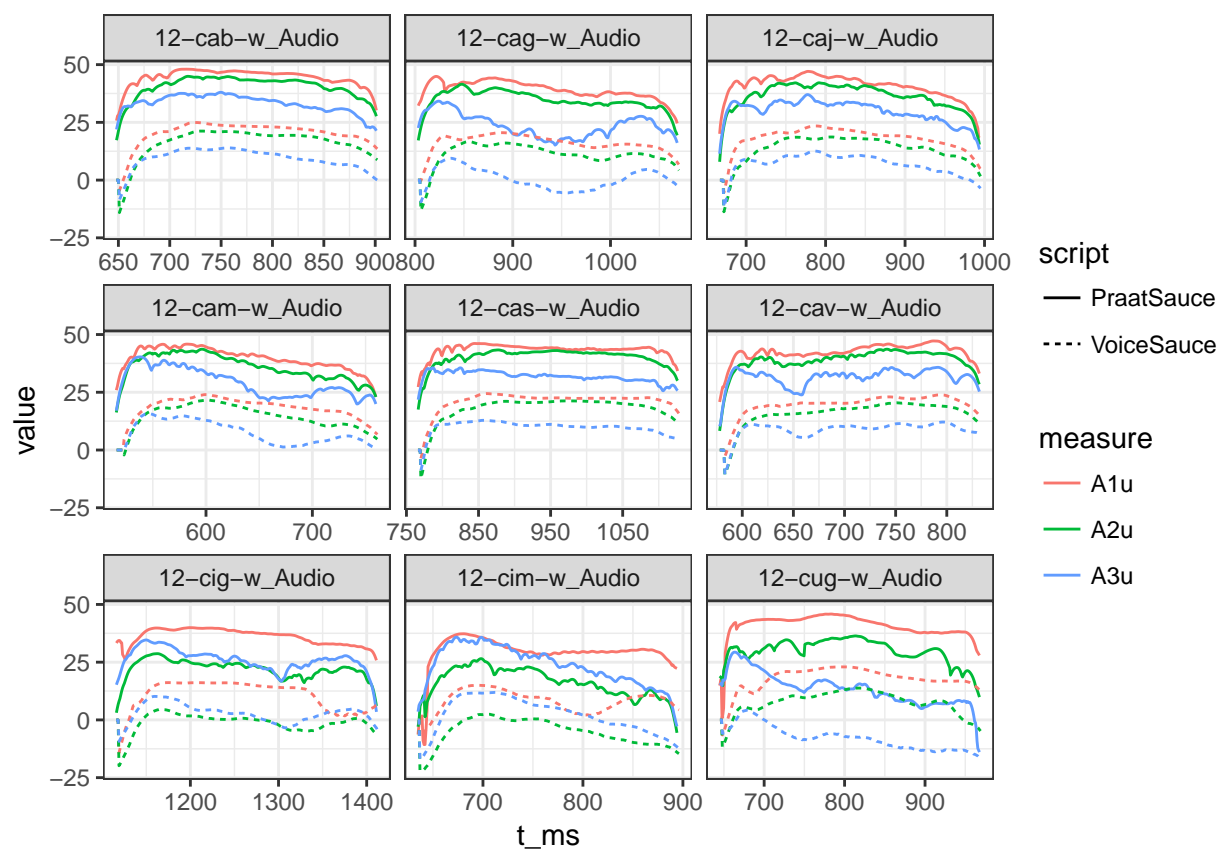
Uncorrected amplitudes

PraatSauce vs. VoiceSauce uncorrected H1, H2, H4

Note that the choice of bandwidth estimator is irrelevant here.



PraatSauce vs. VoiceSauce uncorrected A1, A2, A3



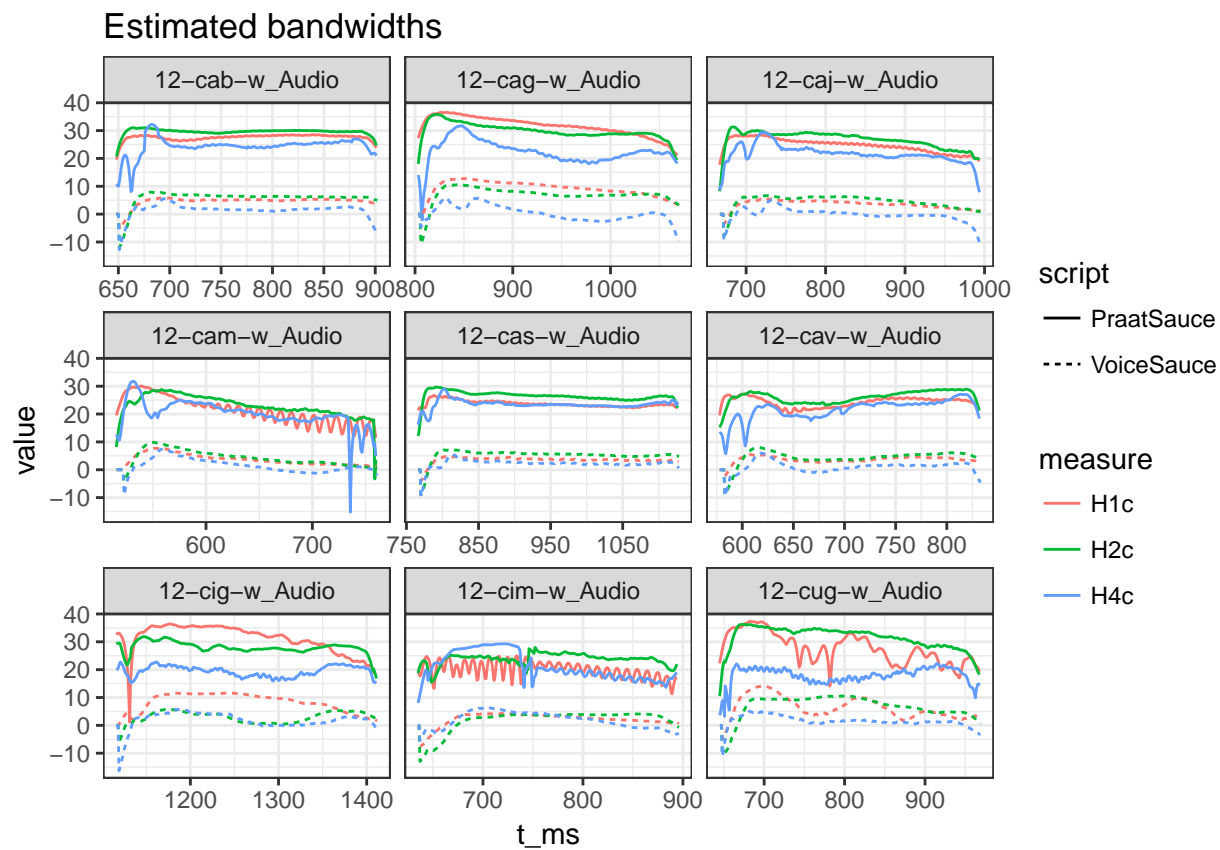
VoiceSauce estimates are consistently 20-25 dB lower than the PraatSauce estimates, and are sometimes negative, which seems...strange. This suggests to me they are being attenuated somewhere, though I have not been able to find the piece of code where this happens.

Corrected amplitudes

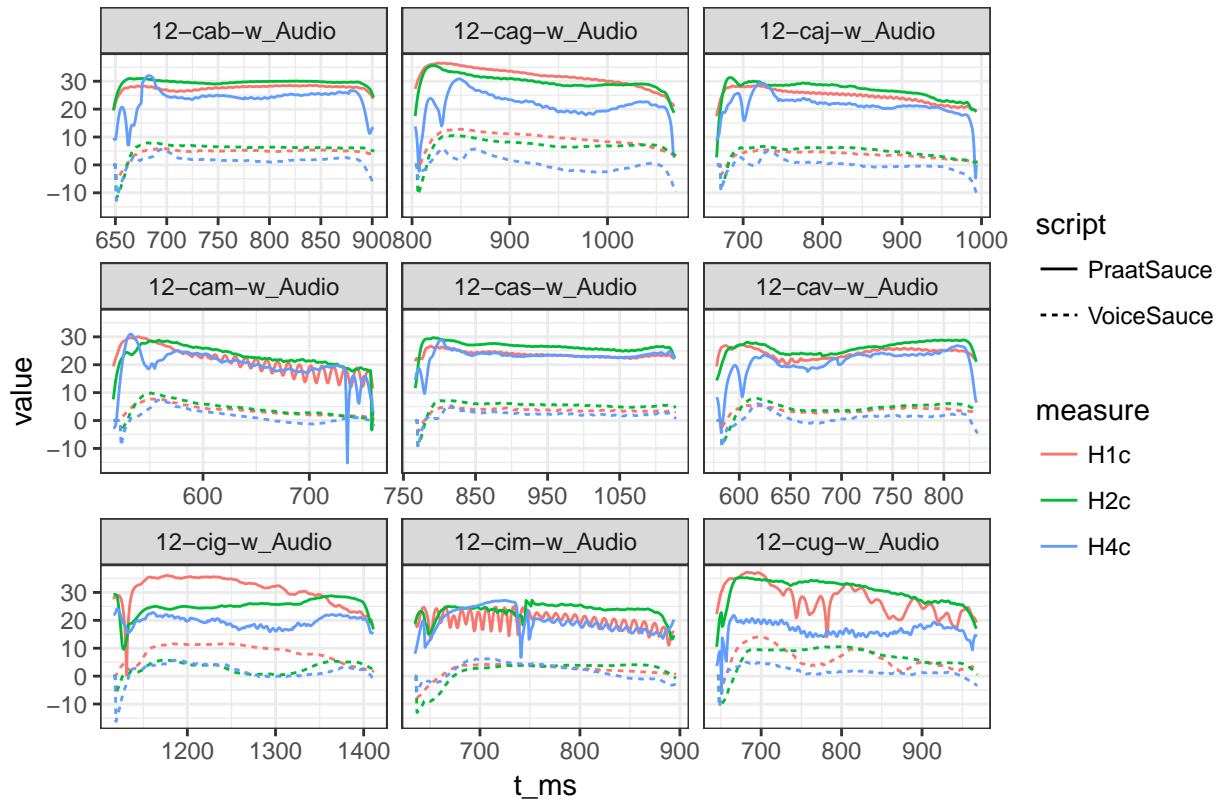
Here, choice of formant bandwidth estimator potentially matters.

In these plots, PraatSauce is using Praat and VoiceSauce is using Snack estimates.

PraatSauce vs. VoiceSauce H1*, H2*, H4*

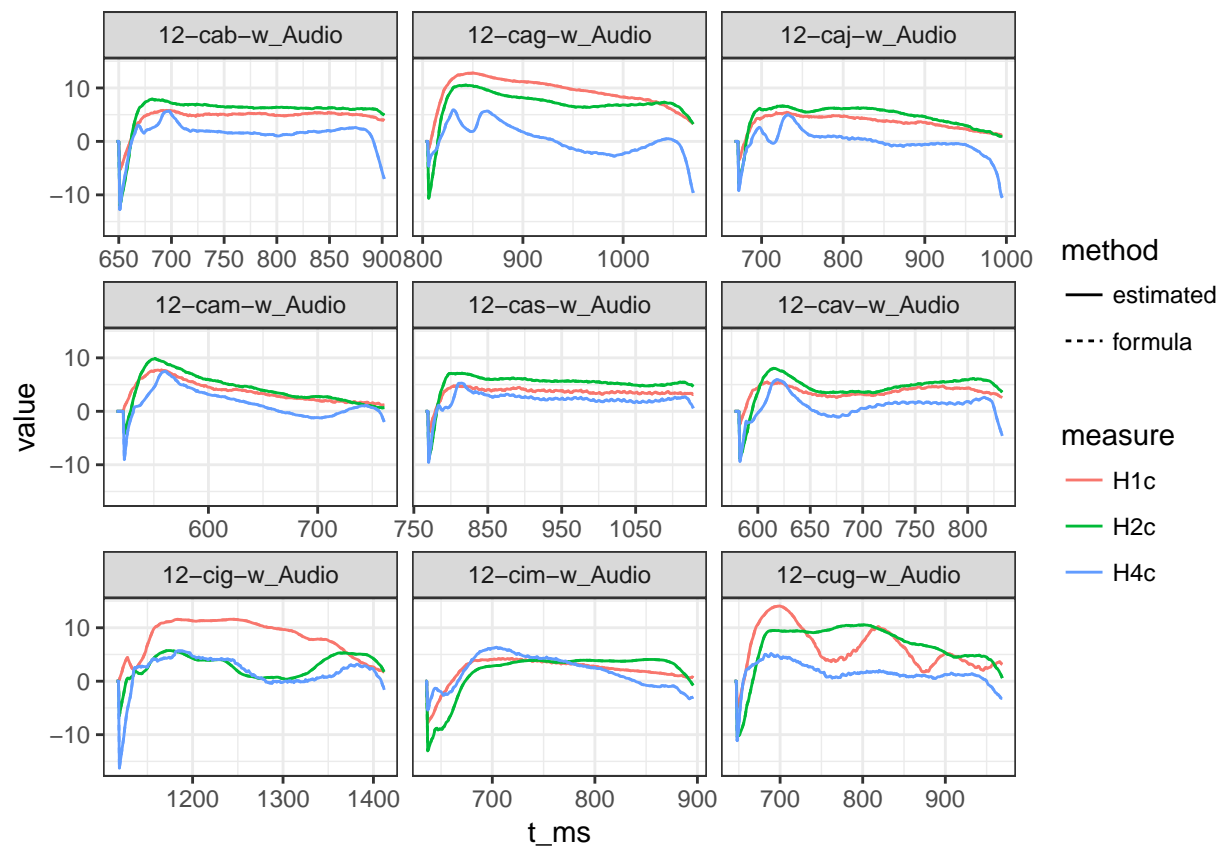


Formula bandwidths



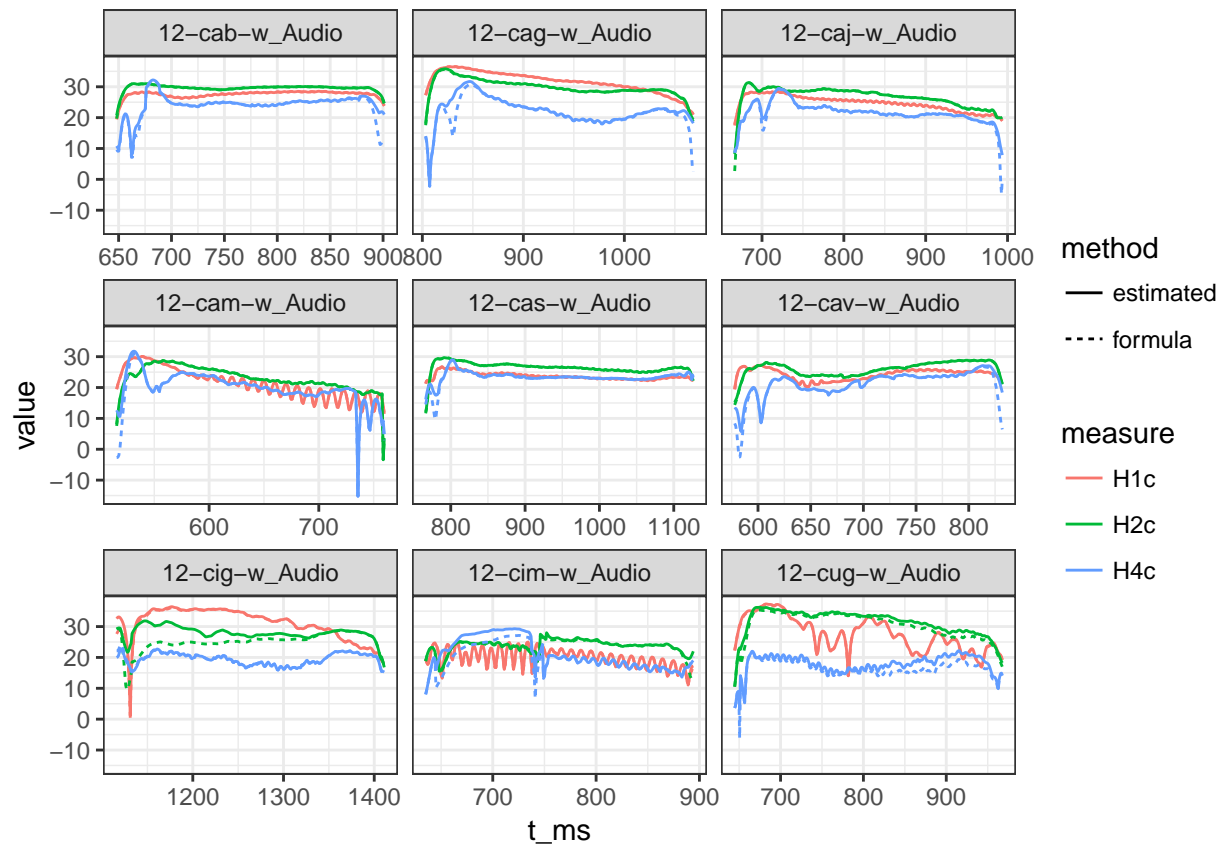
For VoiceSauce, using estimated bandwidths is virtually unnoticeable:

VoiceSauce estimated vs. formula bandwidths, H1*, H2*, H4*

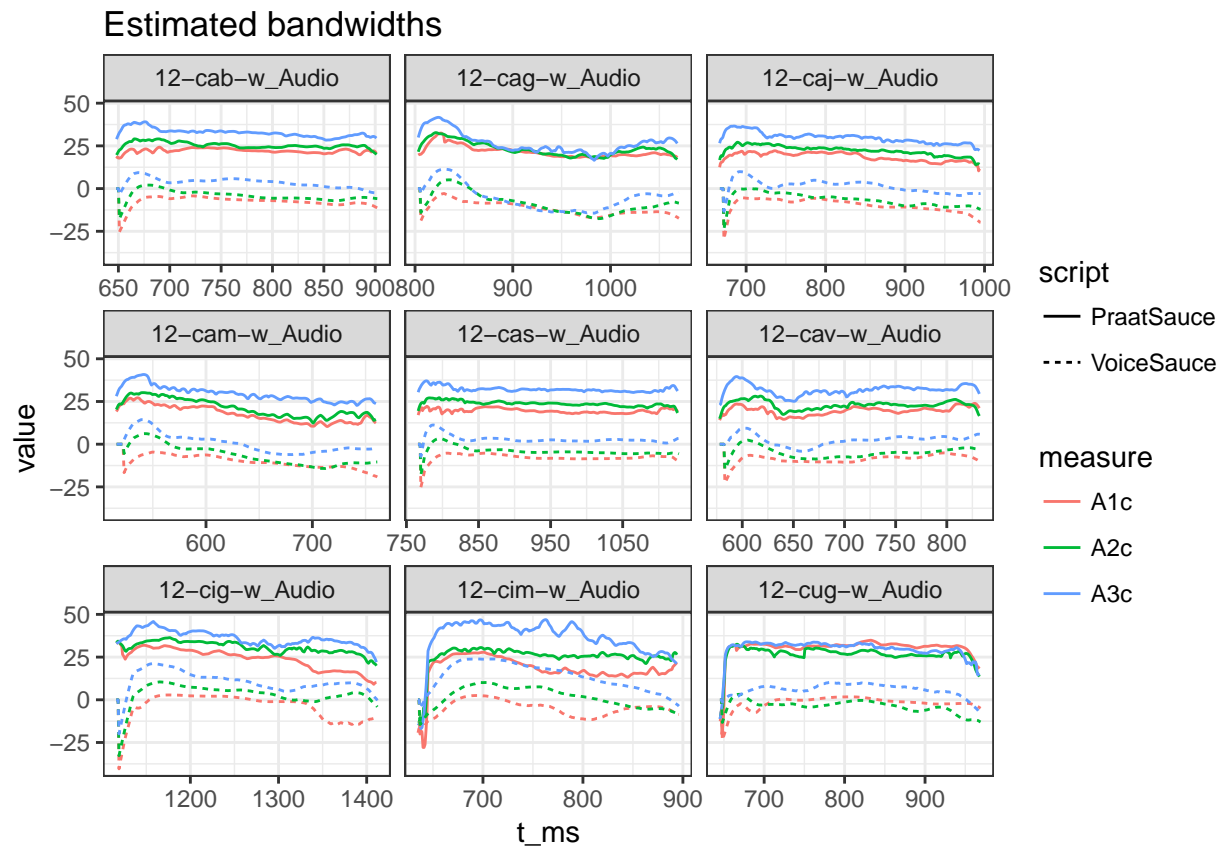


For PraatSauce, using the formula bandwidths makes only very minor differences:

PraatSauce estimated vs. formula bandwidths, H1*, H2*, H4*

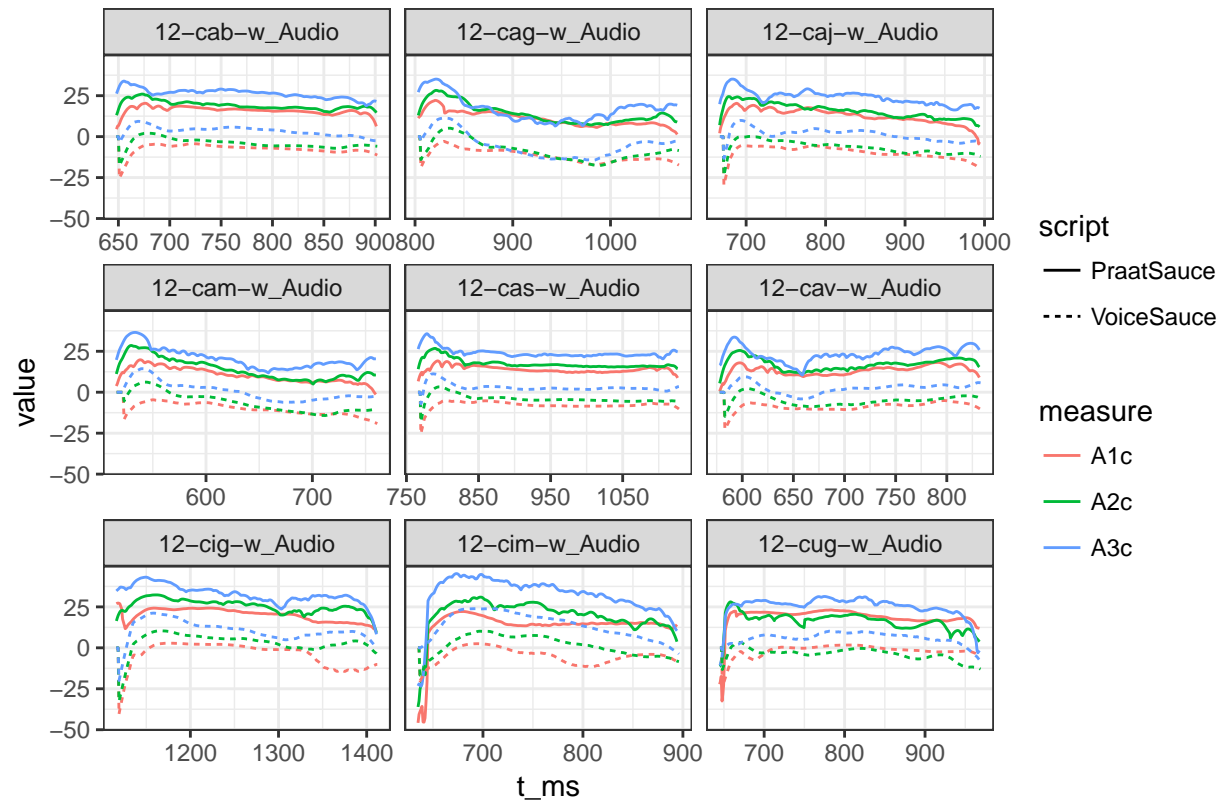


PraatSauce vs. VoiceSauce A1*, A2*, A3*

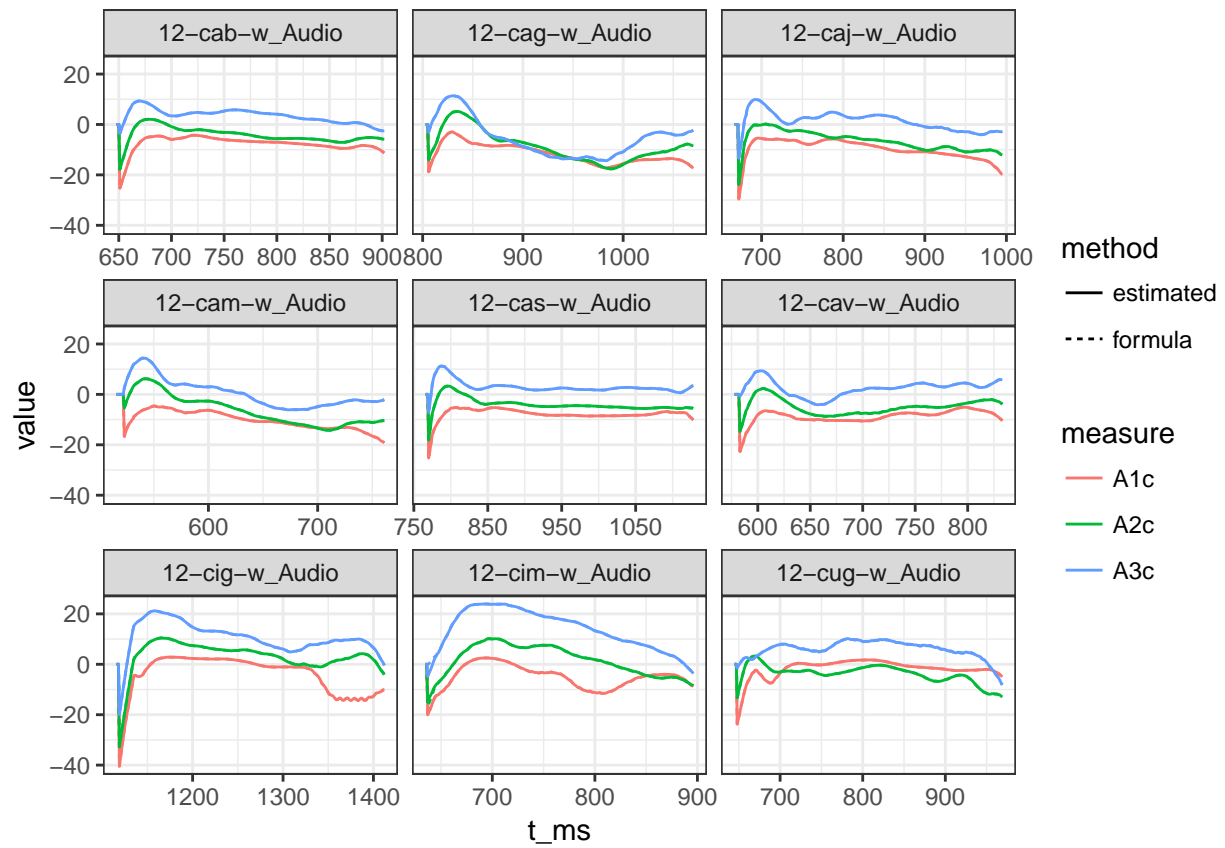


PraatSauce vs. VoiceSauce A1*, A2*, A3*

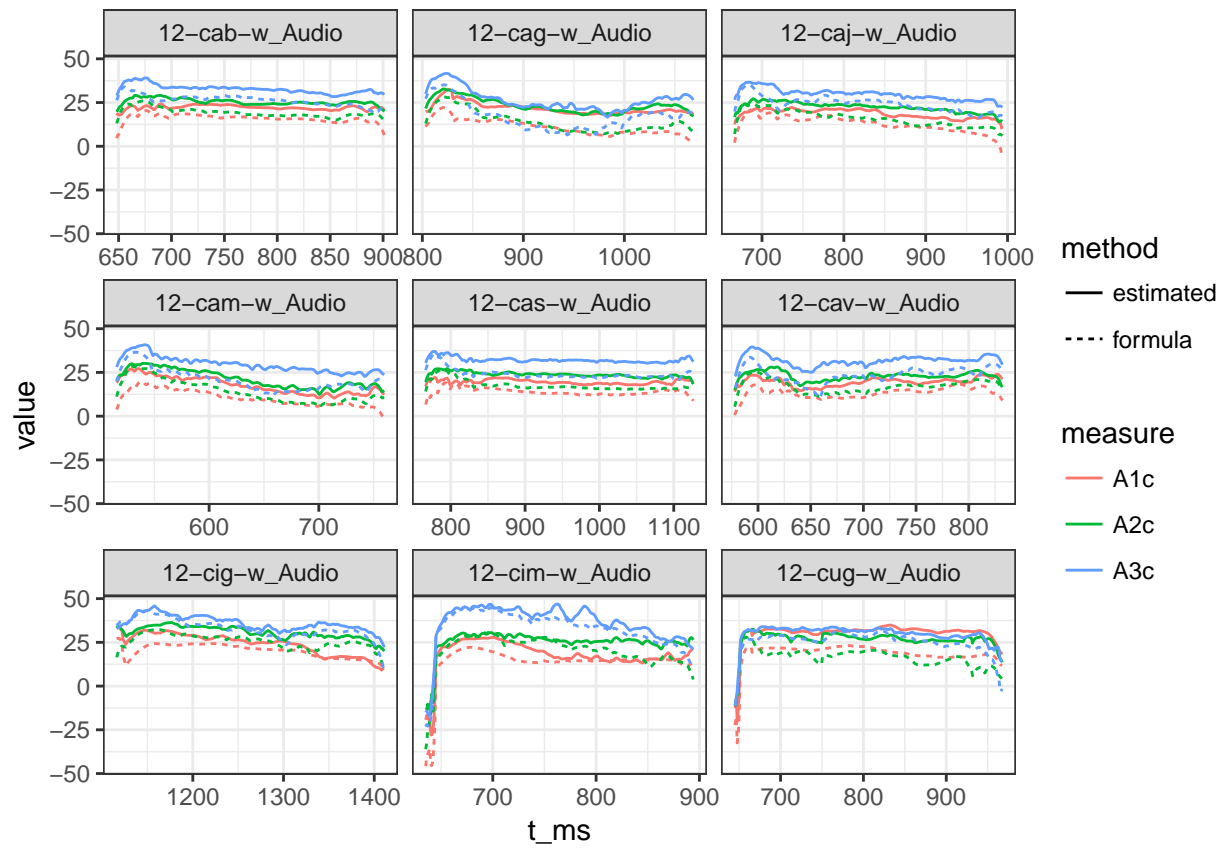
Formula bandwidths



VoiceSauce estimated vs. formula bandwidths, A1*, A2*, A3*

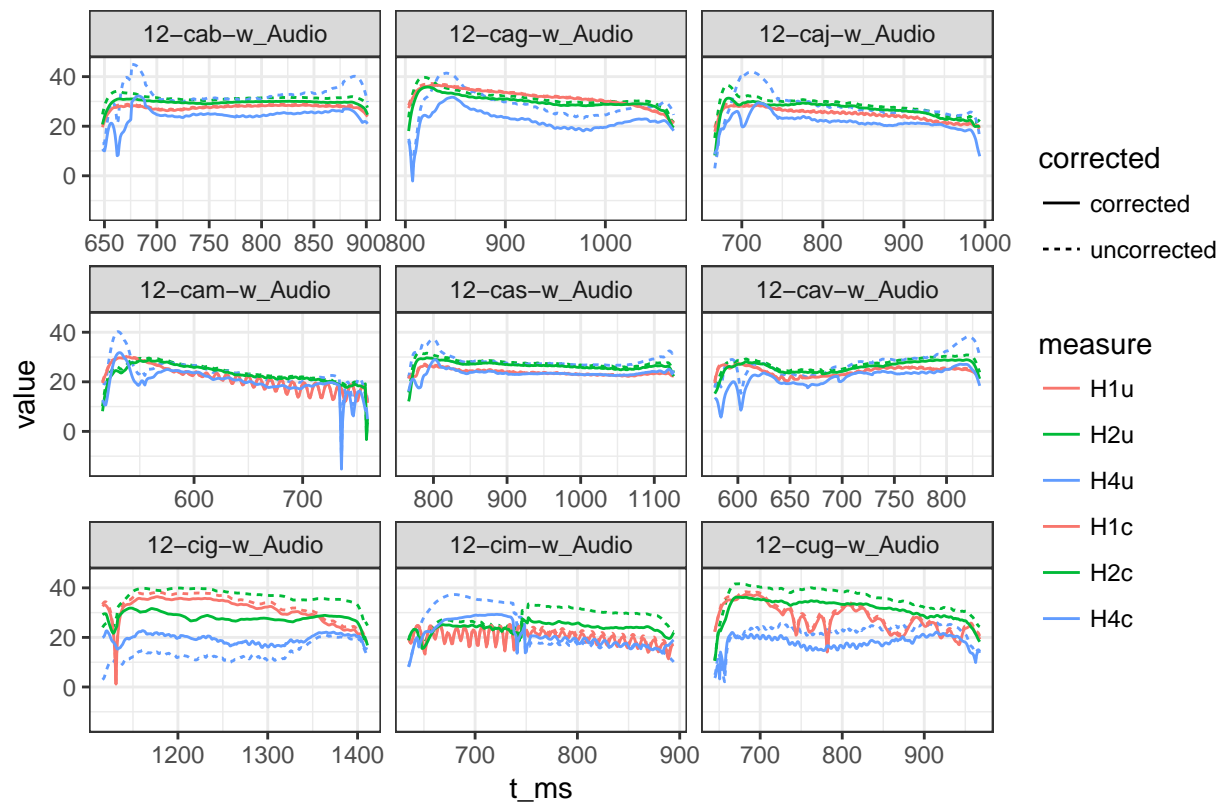


PraatSauce estimated vs. formula bandwidths, A1*, A2*, A3*

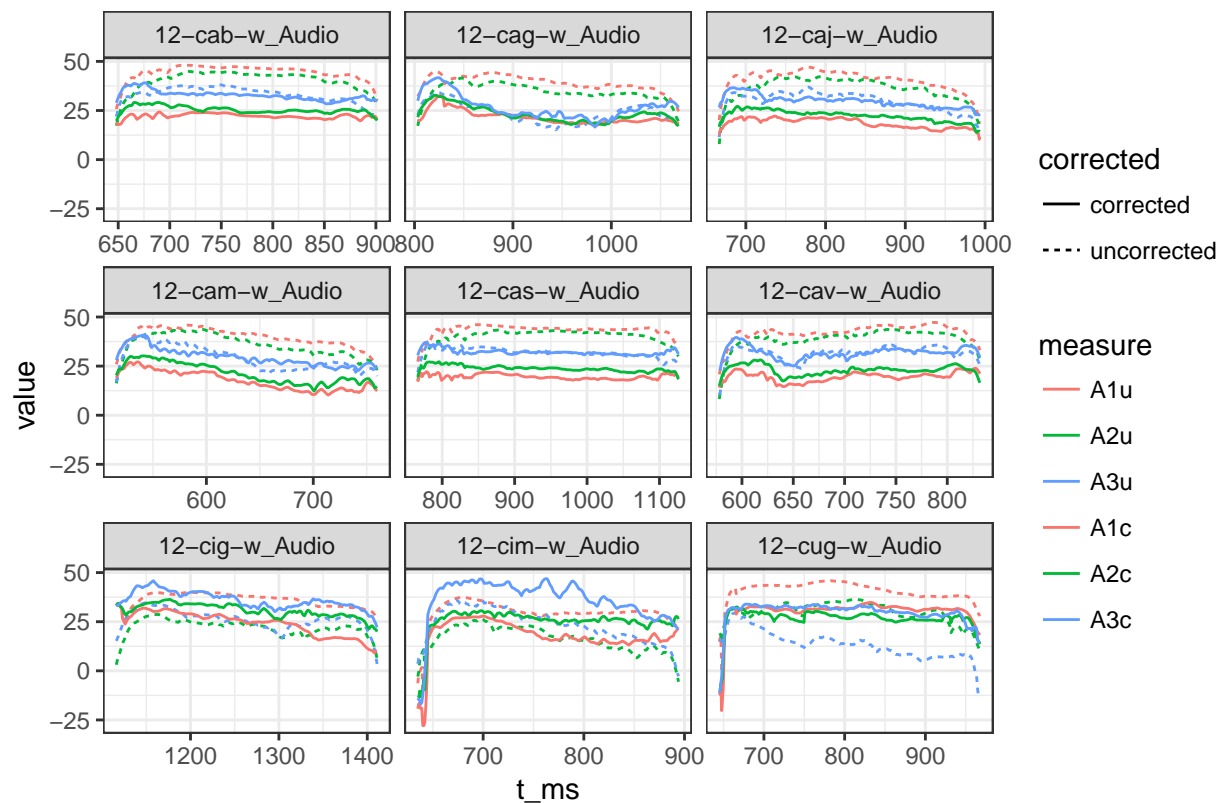


PraatSauce corrected vs. uncorrected

Estimated bandwidths

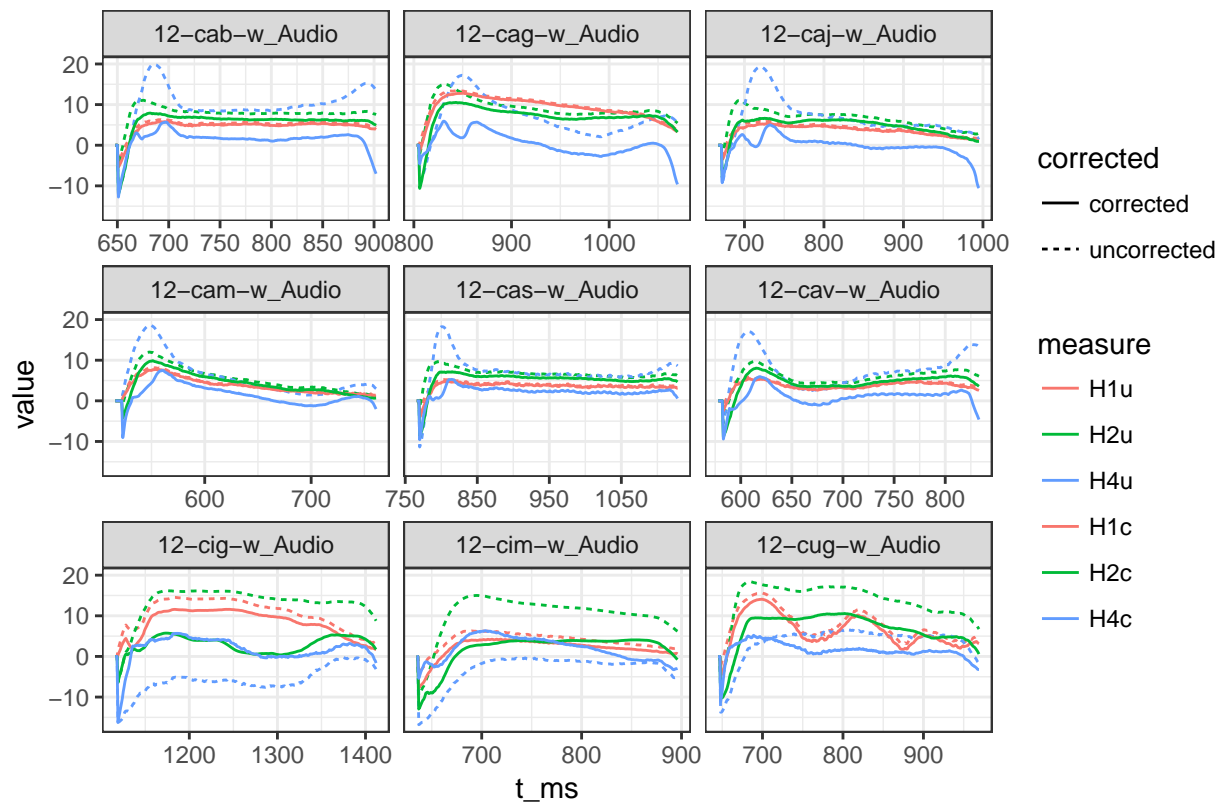


Formula bandwidths

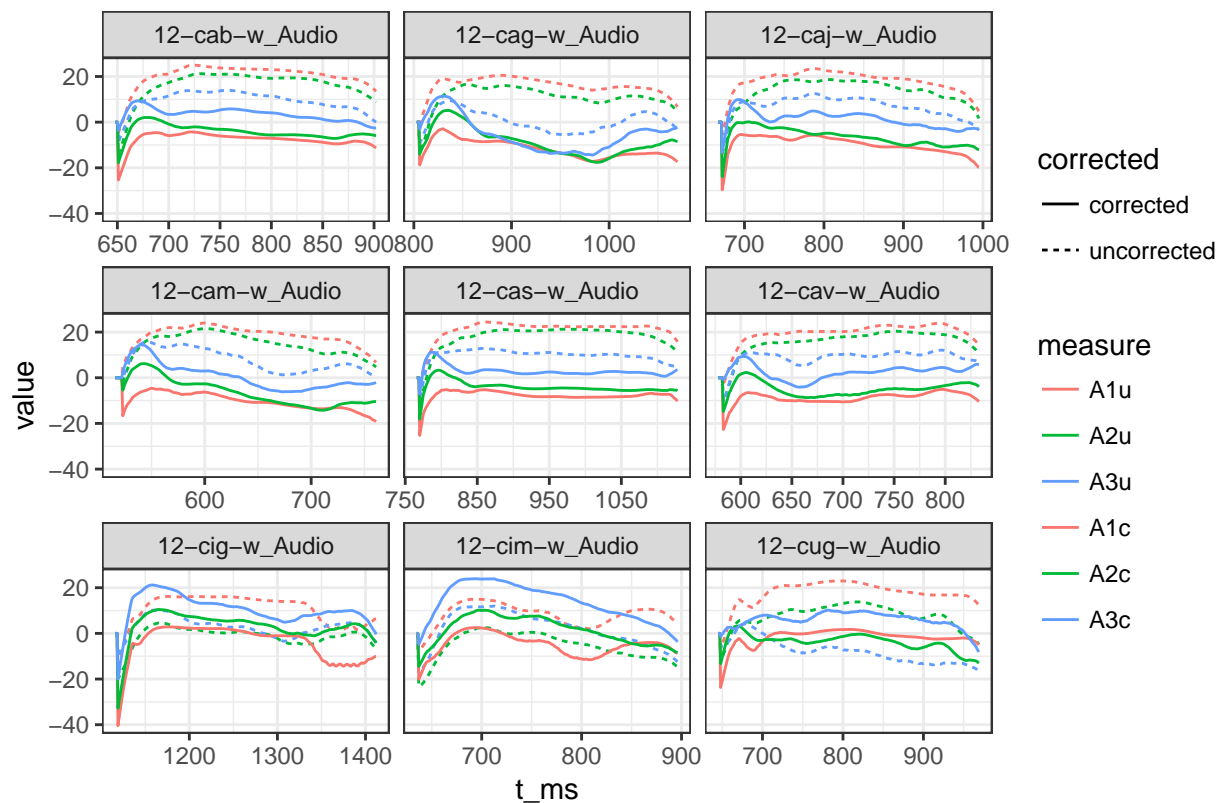


VoiceSauce corrected vs. uncorrected

Estimated bandwidths



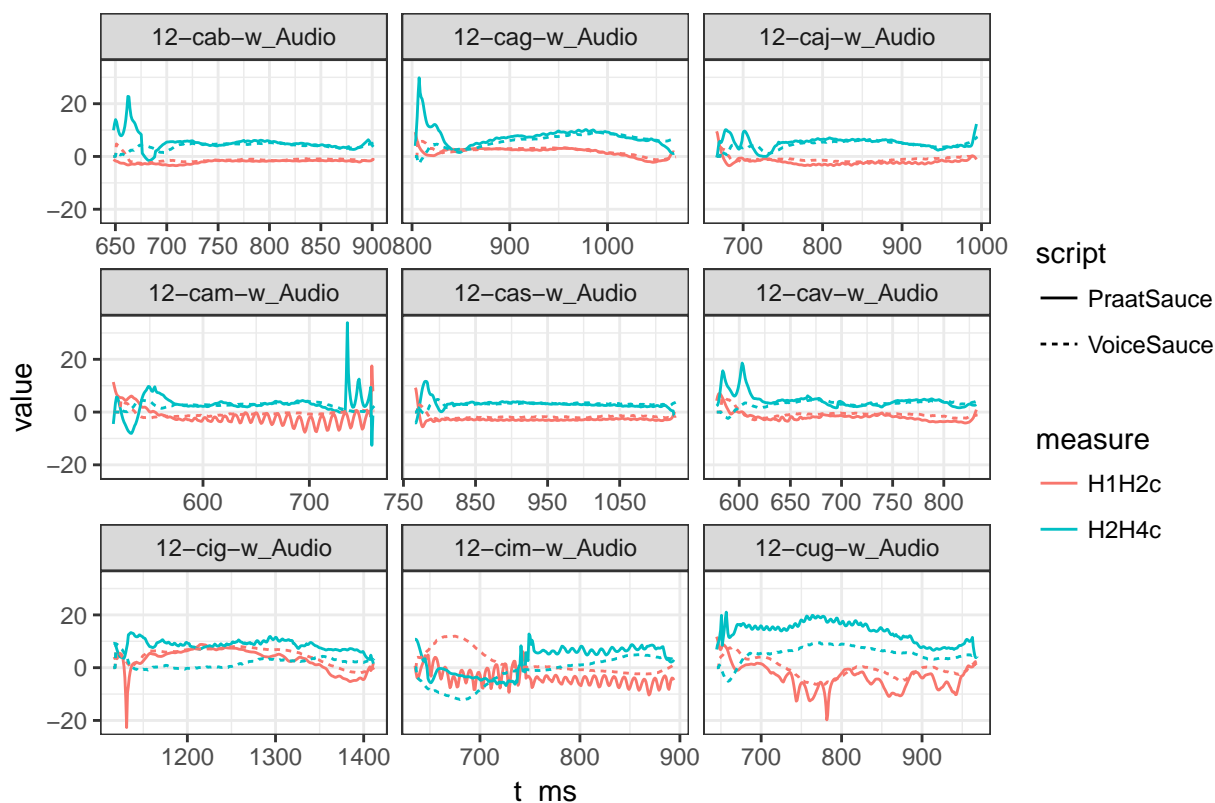
Formula bandwidths



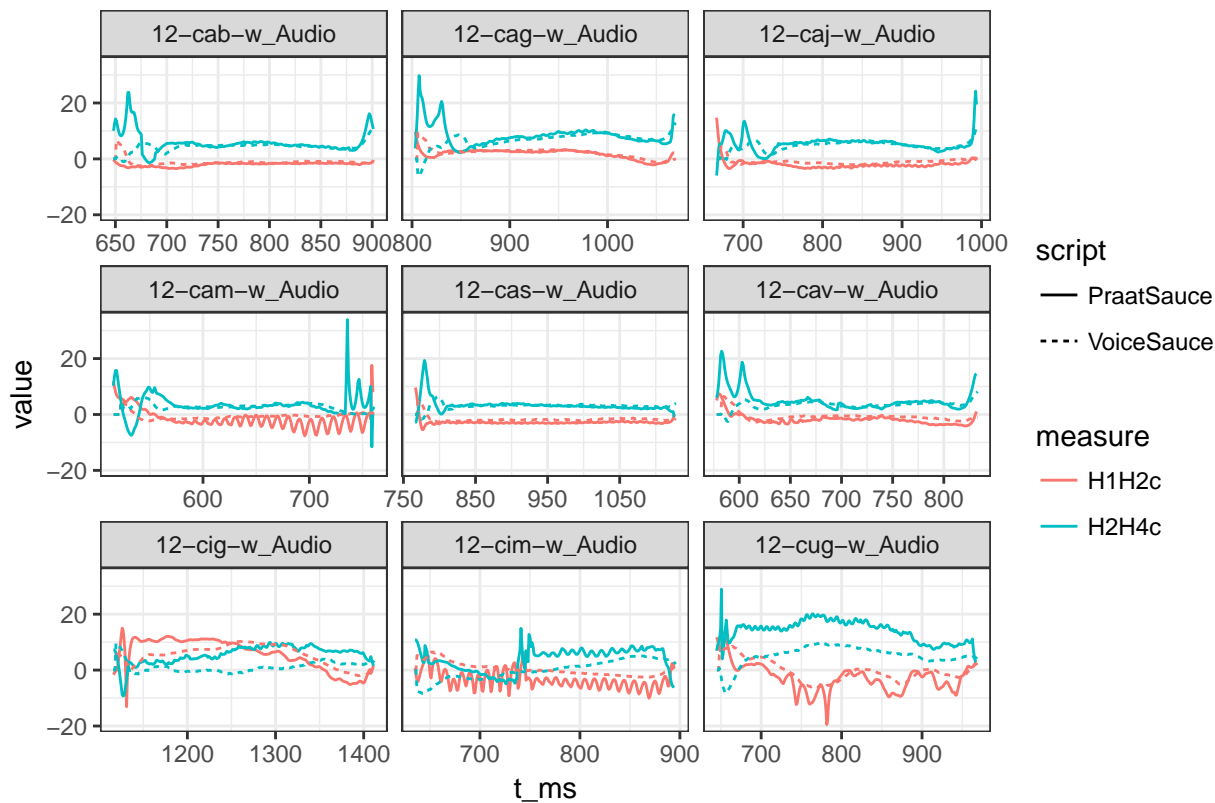
Corrected differences

PraatSauce vs. VoiceSauce $H1^*-H2^*$ & $H2^*-H4^*$

Estimated bandwidths

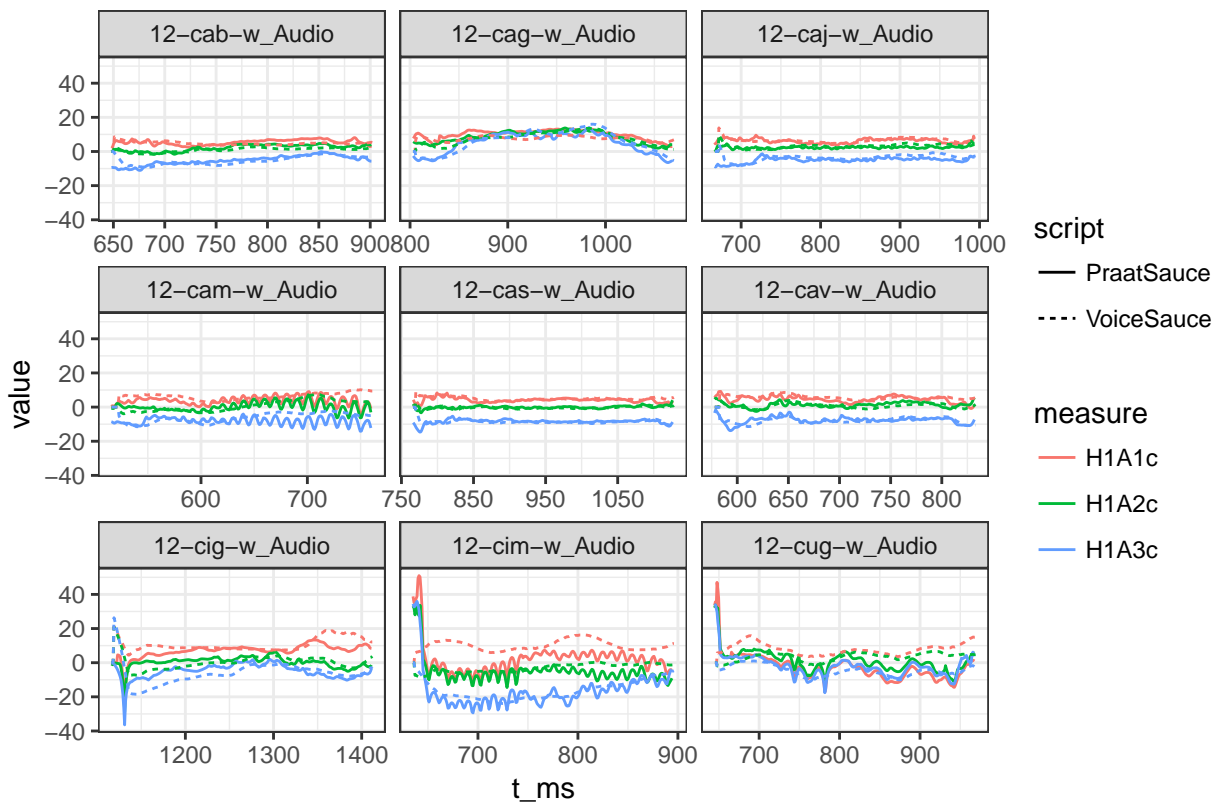


Formula bandwidths

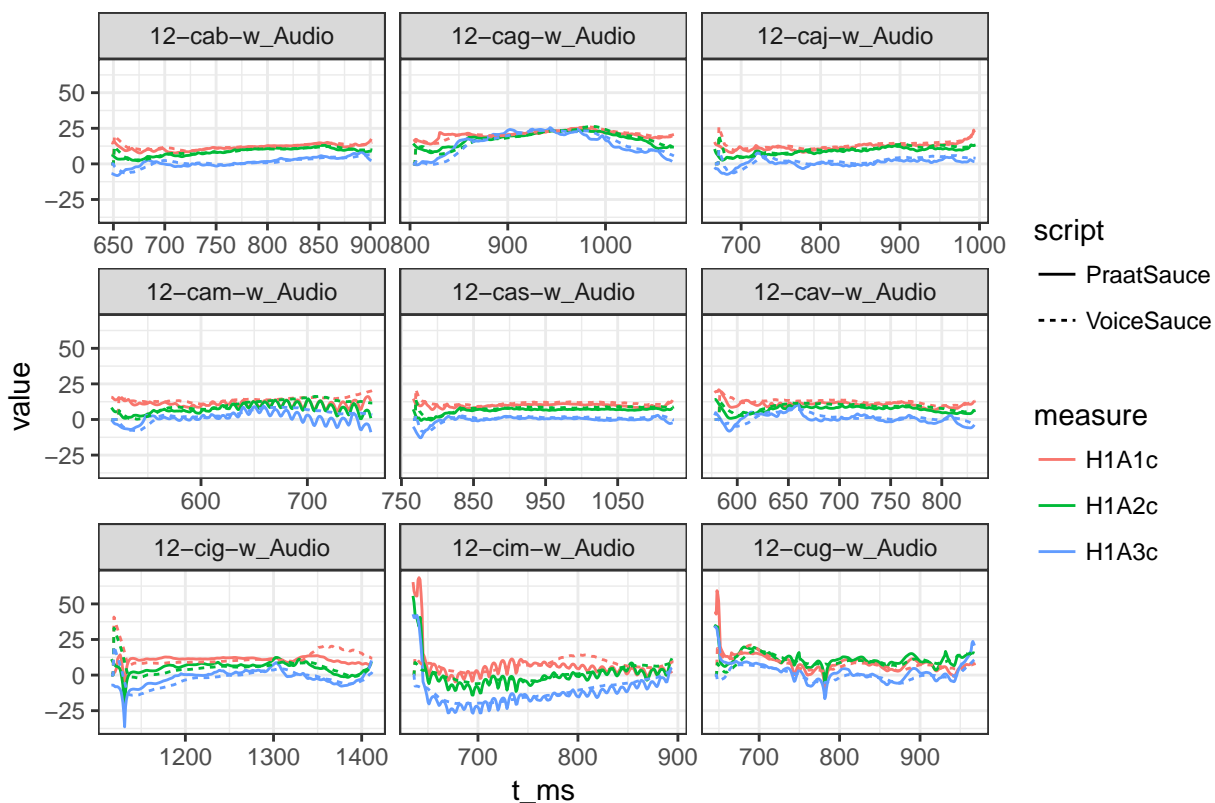


PraatSauce vs. VoiceSauce H1*-A1*, A2*, A3*

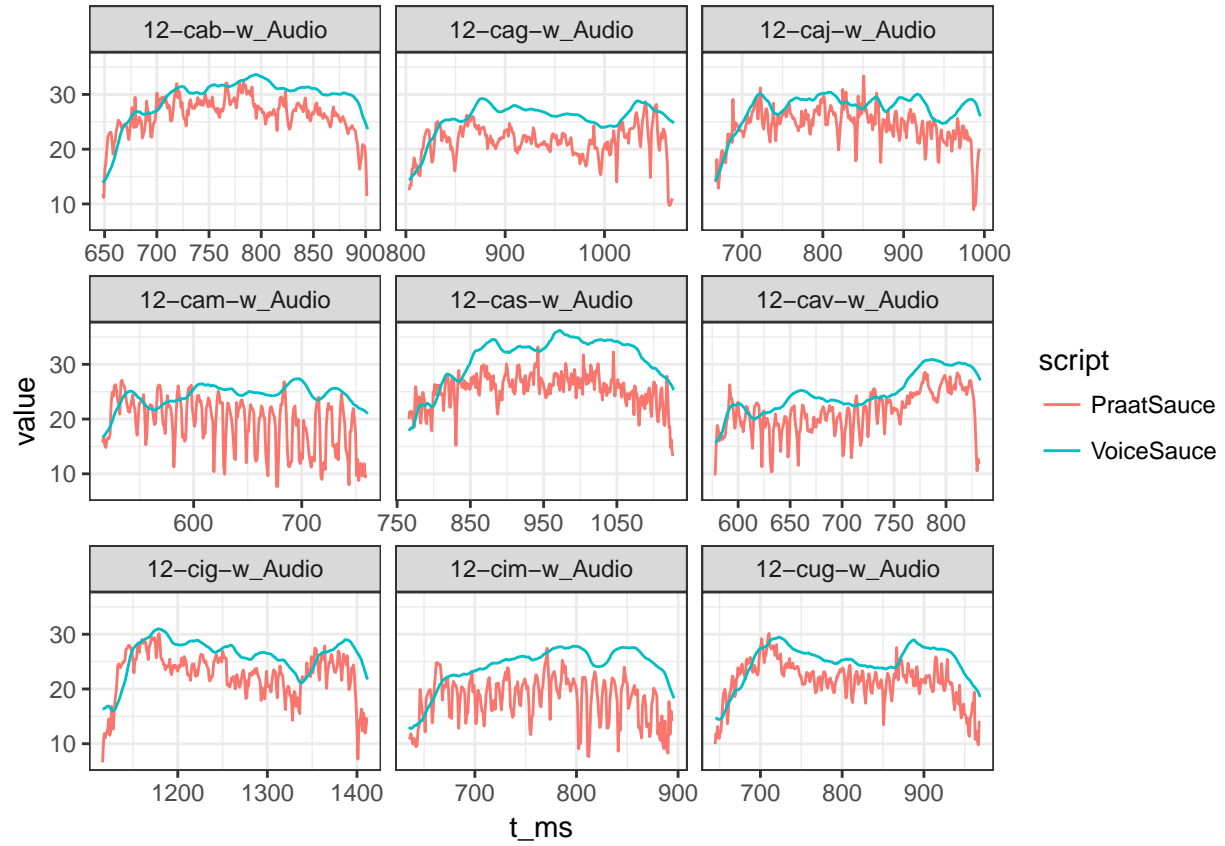
Estimated bandwidths



Formula bandwidths



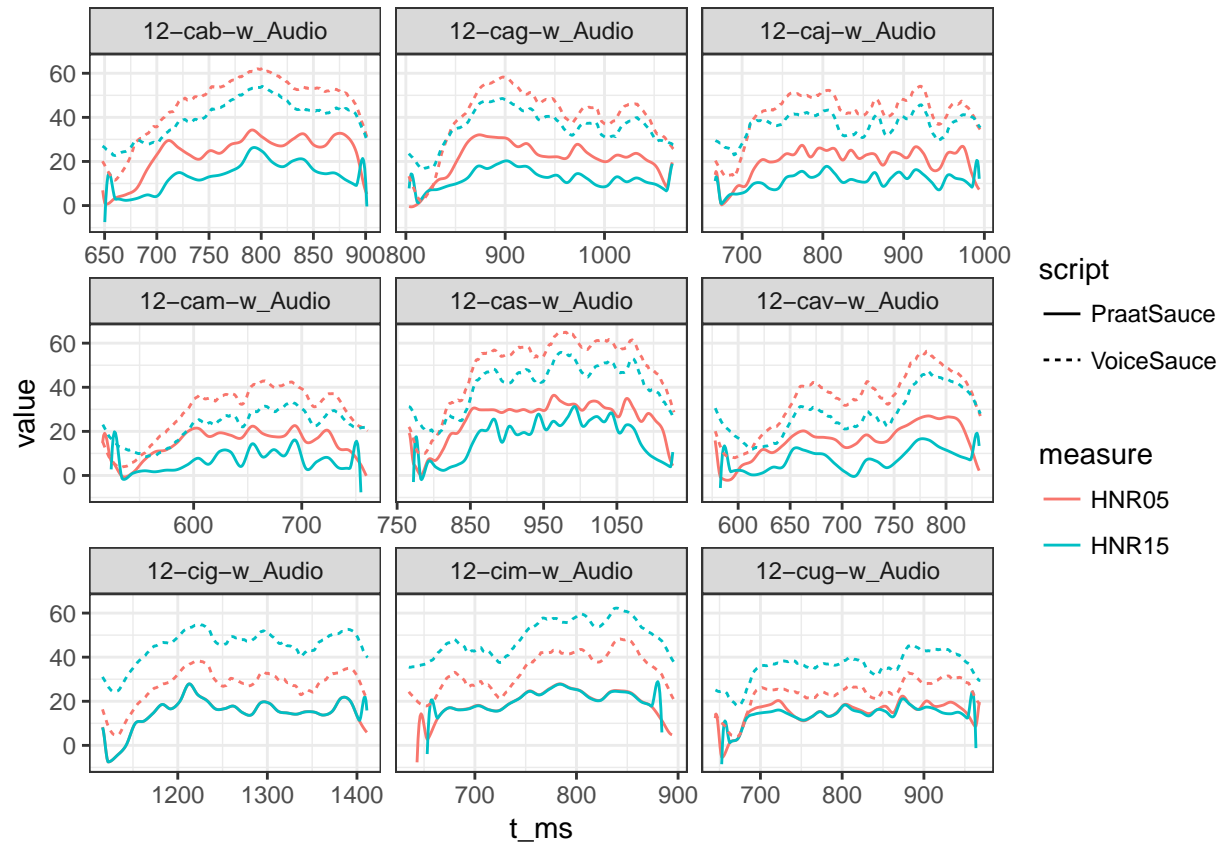
Cepstral peak prominence



Praat(Sauce) estimates are (roughly) comparable if smoothed.

Harmonic to noise ratios

Here just showing HNR05 and HNR15 for clarity.

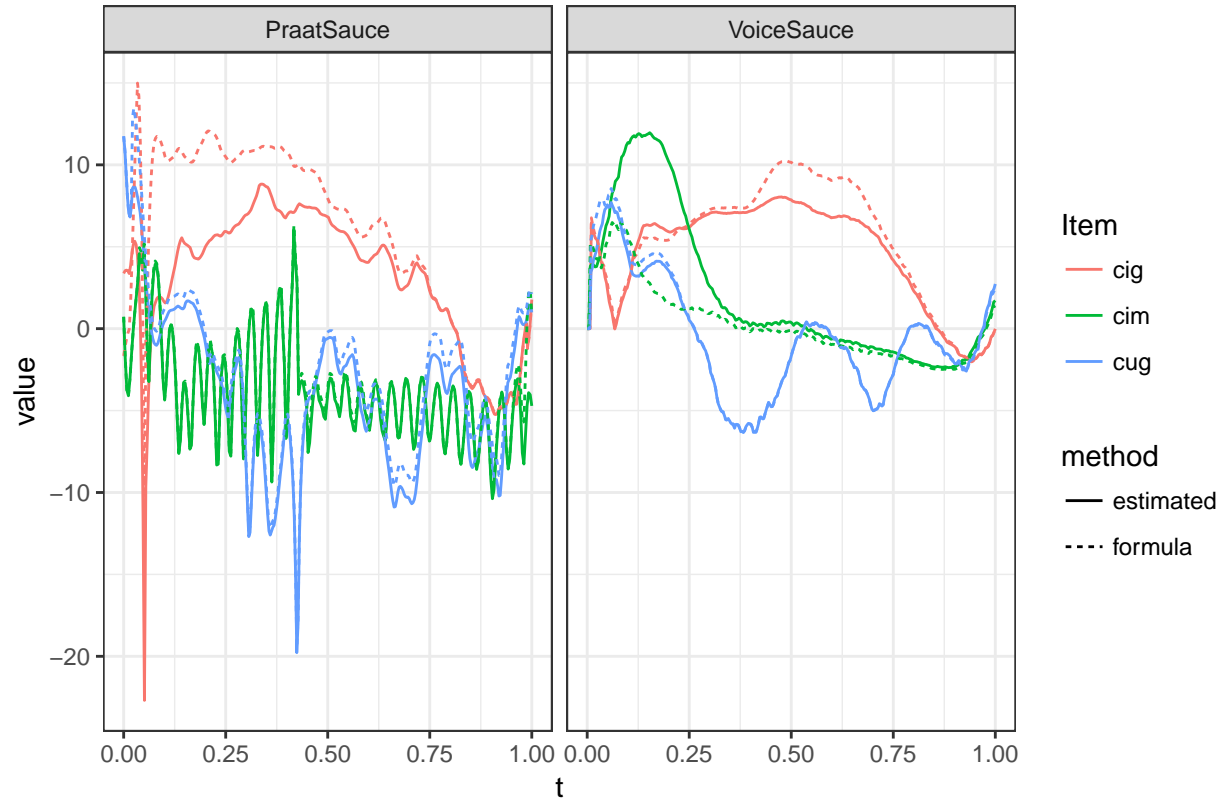


Again, the Praat estimates differ in amplitude, but maintain roughly the same trajectories. However, the PraatSauce implementation is much less sophisticated than that of VoiceSauce, and relies entirely on Praat's `To Harmonicity...` function.

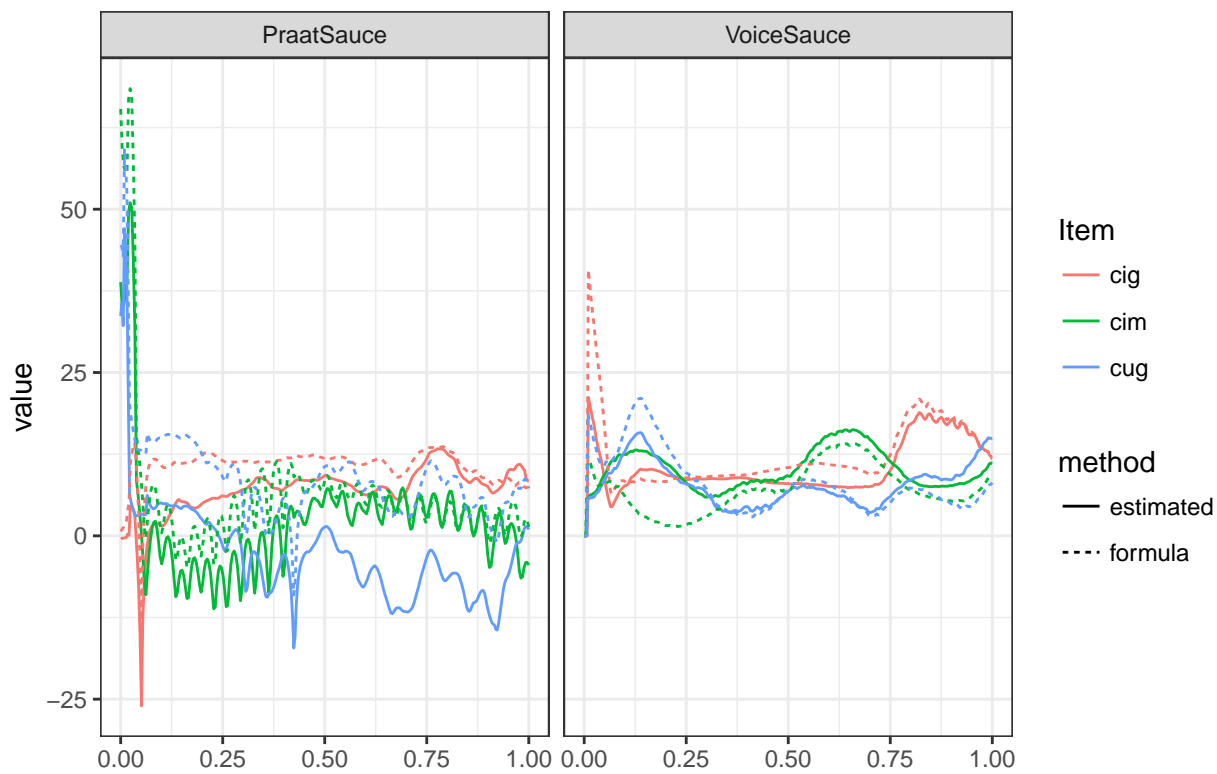
Distinguishing voice qualities

High vowels

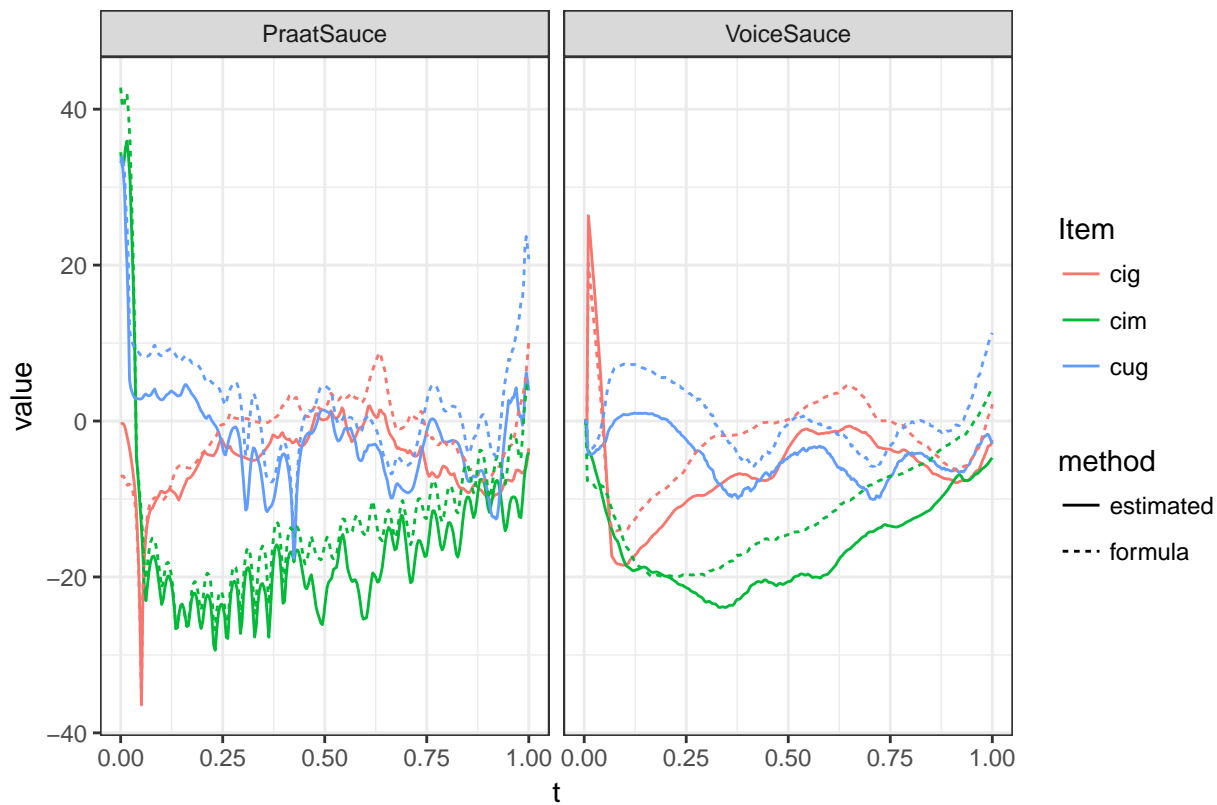
VoiceSauce vs. PraatSauce H1*–H2*



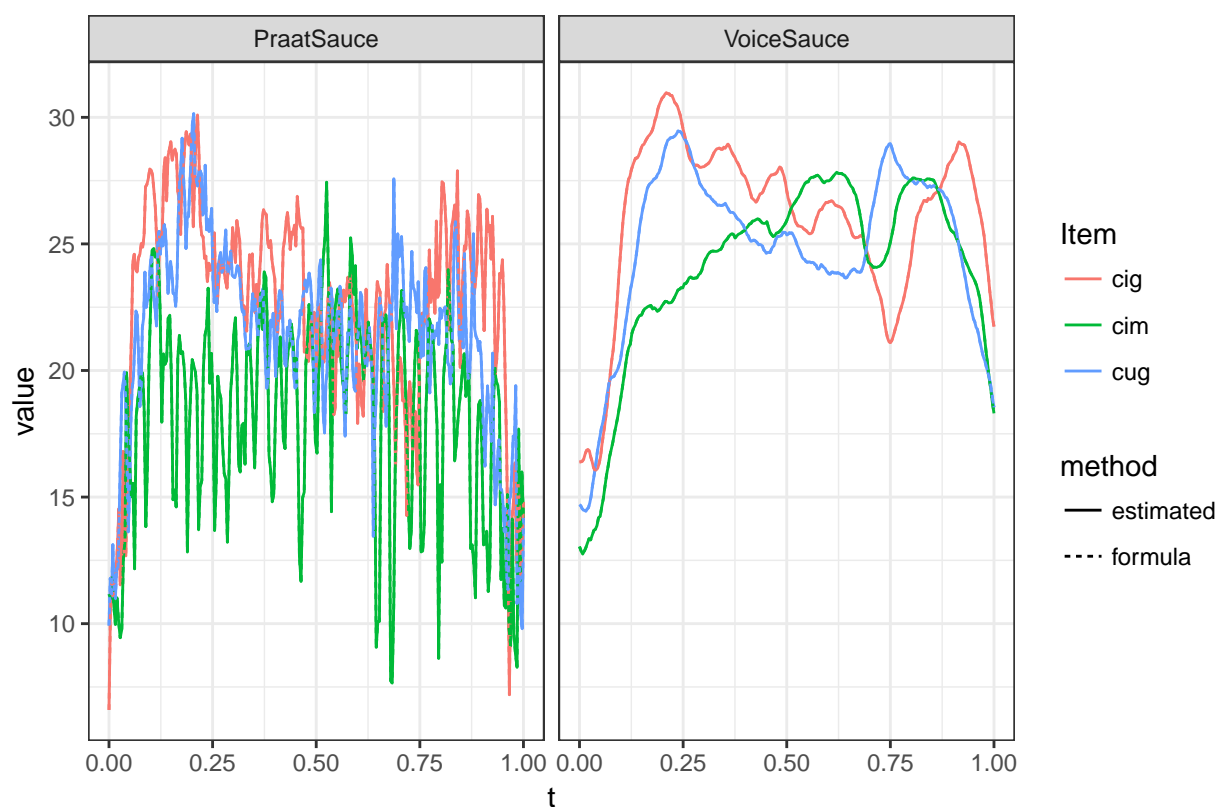
VoiceSauce vs. PraatSauce H1*–A1*



VoiceSauce vs. PraatSauce H1*–A3*



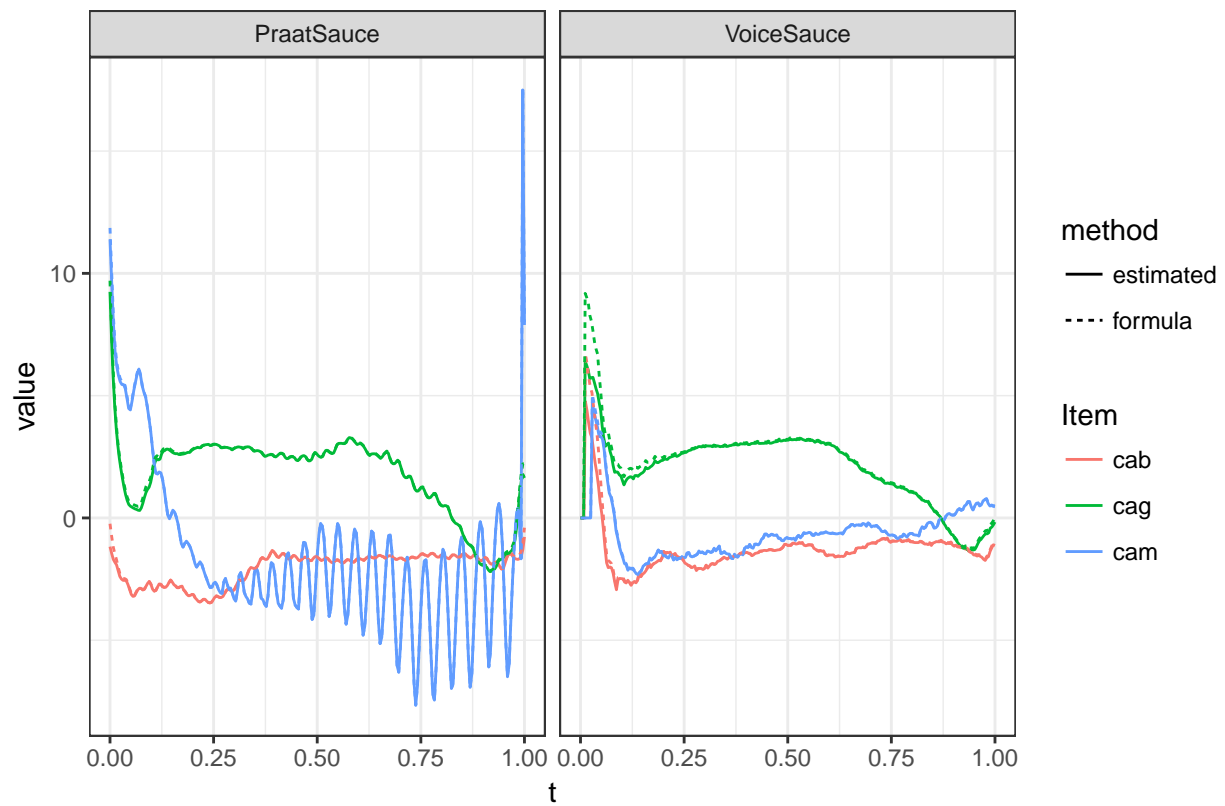
VoiceSauce vs. PraatSauce CPP



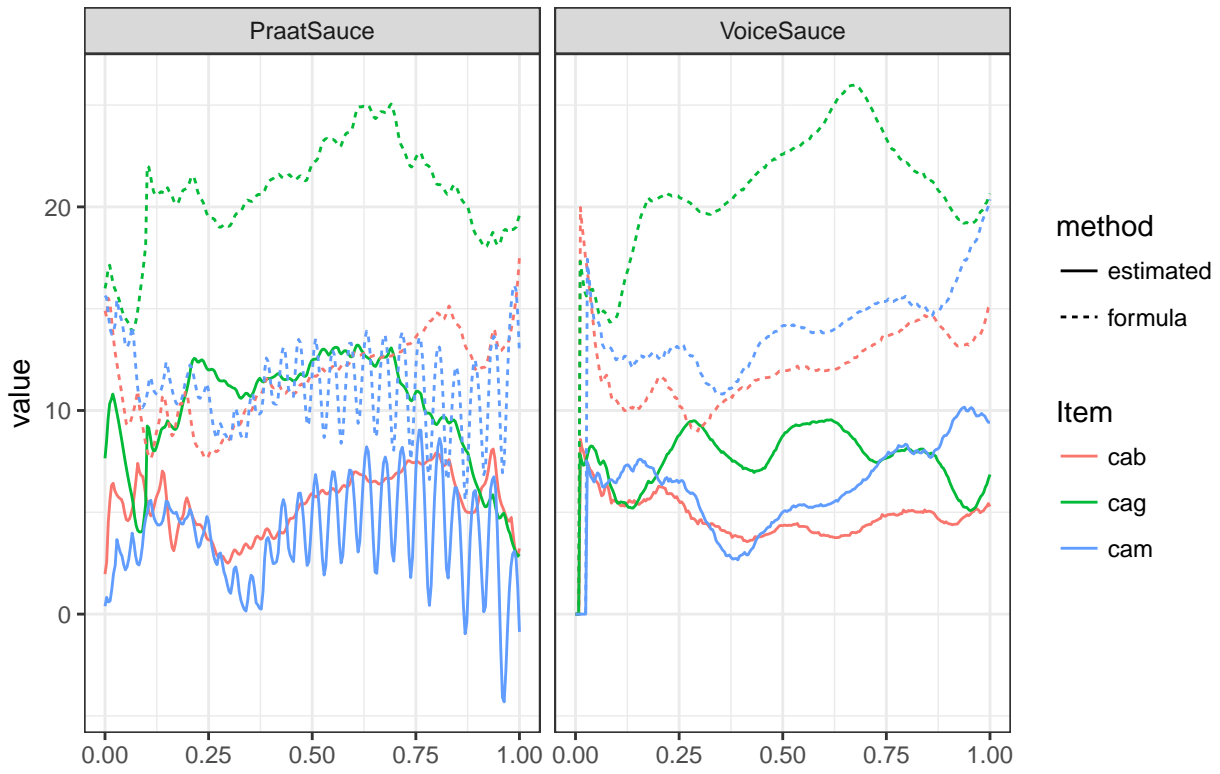
It would appear that neither procedure is correctly diagnosing *cug*, but inspection of the original audio recording suggests that this token is not realized with especially breathy voice.

Low vowels

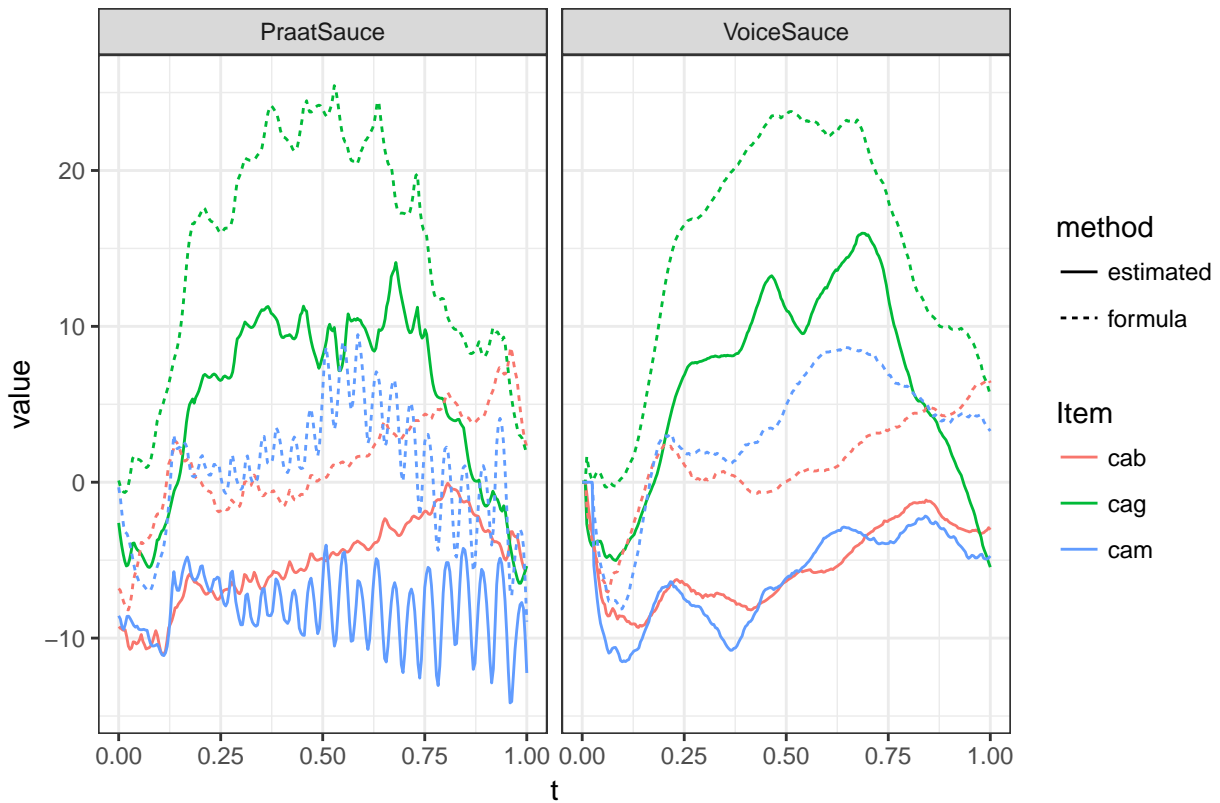
VoiceSauce vs. PraatSauce H1*-H2*



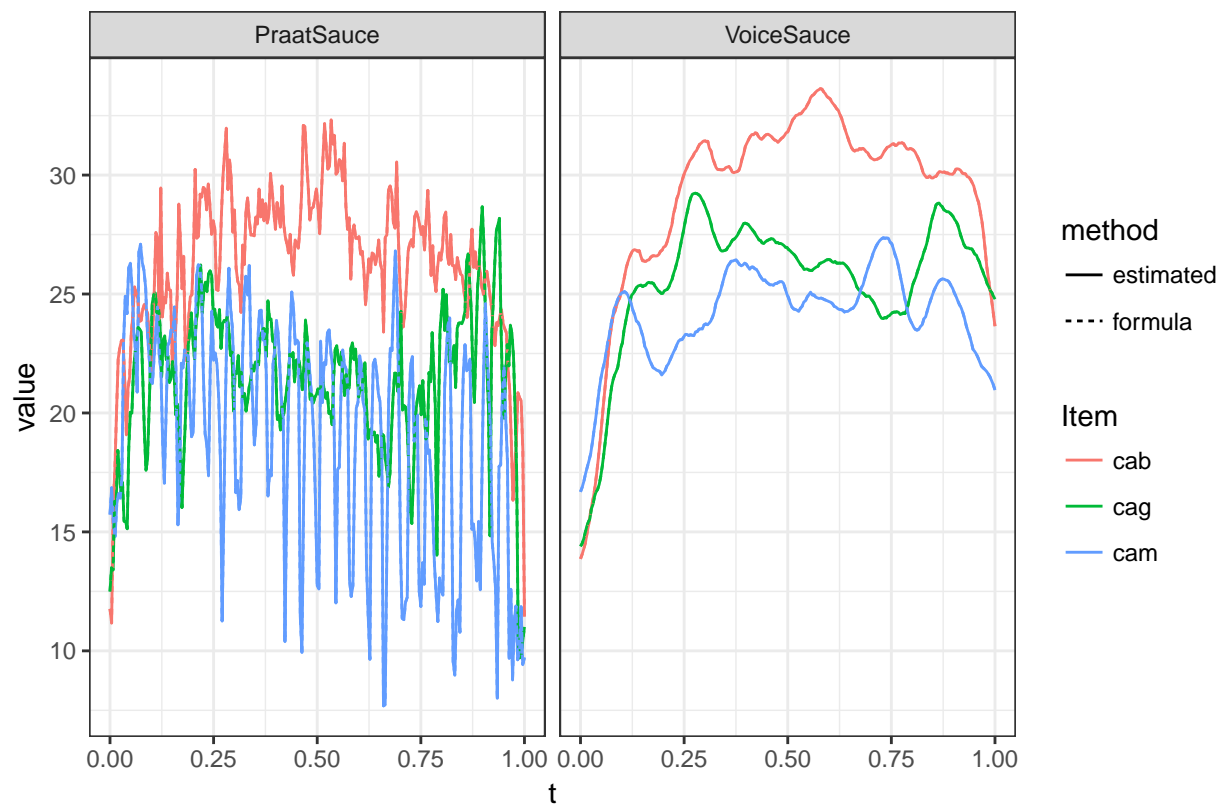
VoiceSauce vs. PraatSauce H1*–A1*



VoiceSauce vs. PraatSauce H1*–A3*



VoiceSauce vs. PraatSauce CPP



PraatSauce CPP values really need to be smoothed/binned.